Development of Numerical Algorithm Based on a Modified Equation of Fluid Motion with Application to Turbomachinery Flow

Von der Fakultät für Ingenieurwissenschaften, Abteilung Maschinenbau der

Universität Duisburg-Essen

zur Erlangung des akademischen Grades

DOKTOR-INGENIEUR

genehmigte Dissertation

von

Bo Wan
aus
Jiangsu, China

Referent: Prof. Dr.-Ing. F.-K. Benra
Korreferent: Prof. Dr. S. H. Sohrab
Tag der mündlichen Prüfung: 03. September 2012

Abstract

On the basis of the scale-invariant theory of statistical mechanics, Sohrab introduced a linear equation termed the “modified equation of fluid motion.” Preliminary investigations have shown that this modified equation can be extended to solve flow problems. Analytical solutions of basic flow problems were derived using this equation. In all cases the match between estimated and experimental data was good. These results stimulated further applications of this modified equation in the development of a CFD code to obtain numerical solutions of turbomachinery flow problems.

In the present work, a novel numerical algorithm based on the aforementioned modified equation has been developed to solve turbomachinery flow problems. In order to avoid dealing with more technical conditions on the scale–invariant form of the energy equation, this investigation is restricted to incompressible flow.

On the basis of the work done by Sohrab, the derivation process of the modified equation for incompressible flow is presented with more emphasis on its linear property as compared to the Navier–Stokes equation for incompressible flow. Furthermore, a detailed analysis of the present discretisation technique for the modified equation is performed. As compared with the Navier–Stokes equation, the numerical errors resulted from the discretisation of the modified equation, including the truncation and discretisation errors are discussed as well as the stability conditions. On the basis of the analysis above, a novel numerical solver is established by using the open source code OpenFOAM. A corresponding iteration technology is determined for the solver to dedicate reliable and efficient solutions to the algebraic linear equations derived from the FVM discretisation of the modified equation. In addition, a dynamic mesh solver developed on the basis of the ALE method is also built up to resolve the transient flow problems. Finally, the developed solver is applied to solve the modified equation for several flow models including the fundamental boundary layer problems, flow around an airfoil, turbulent secondary flow in a curved duct and cascades, and rotor–stator interactions in radial pumps. The results are evaluated by comparing them with those obtained by other methods, including the numerical results from a Navier–Stokes solver and measured data. For all investigated cases the computational effort and the accuracy of the solver are emphasized. The comparisons indicate that the developed solver, which is based on the modified equation of fluid motion, requires less than half the computation time of the Navier–Stokes solver, and it produces physically reasonable results validated by measured data.

The encouraging result demonstrates the scientific credibility of the developed solver for application to turbomachinery flows. Further, if it is extended to design processes in the turbomachinery industry, significant improvements in the design cycle cost can be reached.

Acknowledgments

I would like to express my gratitude to all those who helped me during my study in Germany.

My sincere gratitude goes first and foremost to Prof. Dr.-Ing. F.-K. Benra, my supervisor, for his inspirational guidance and constant encouragement. He offered me the opportunity to start this interesting topic, and gave me great help by providing me with necessary materials, advice of great value and inspiration of new ideas during my research. I am extremely grateful for his support in the accomplishment of my study in Germany.

High tribute shall be paid to Prof. Siavash H. Sohrab, Northwestern University, IL, who introduced the scale-invariant form of the equation of motion, for his instructive theoretical support to my research and the review of my dissertation.

I am also deeply indebted to all my colleagues and student assistants in the chair for Turbomachinery, University of Duisburg-Essen. Special thanks should go to Dr.-Ing. H. J. Dohmen who has given me valuable suggestions and patient guidance.

Last but not the least, my gratitude also extends to my family who has been assisting, supporting and caring for me throughout my life.

Bo Wan
Duisburg, October 2012

Contents

1 INTRODUCTION 1

1.1 State of the Art of the Modified Equation of Fluid Motion and Its Applications....1

1.2 Motivations.....3

1.3 Present Contributions.....3

1.4 Thesis Structure.....4

2 MODIFIED EQUATION OF FLUID MOTION..... 5

2.1 Scale–invariant Model of Statistical Mechanics.....5

2.2 Convective, Diffusion and Local Velocities.....8

2.3 Modified Equation of Fluid Motion for Incompressible Flow.....10

2.3.1 Derivation of the Modified Equation of Fluid Motion..... 10

2.3.2 Relation between the Modified Equation and the Navier–Stokes Equation..... 13

2.4 Reynolds Averaged Modified Equation of Fluid Motion.....14

3 NUMERICAL ERROR 17

3.1 Taylor Series Expansion.....18

3.2 Truncation Error.....20

3.3 Discretisation Error – Numerical Diffusion.....22

4 DEVELOPED SOLVER..... 28

4.1 1D Discretisation.....29

4.1.1 CD Discretisation..... 30

4.1.2 UD Discretisation..... 30

4.2 Solver Algorithm & Iteration.....32

4.2.1 Navier–Stokes Solver..... 32

4.2.2 Sohrab Solver..... 35

4.3 Dynamic Mesh Solver.....37

4.3.1 Navier–Stokes Solver..... 37

4.3.2 Sohrab Solver 40

5 APPLICATIONS OF THE SOHRAB SOLVER 42

5.1 BOUNDARY LAYER FLOW OVER A FLAT PLATE 42

5.1.1 Laminar Flow 43

5.1.1.1 Analytical Solution 43

5.1.1.2 Numerical Solution 44

5.1.1.3 Results & Comparisons 47

5.1.1.4 Numerical Error Comparisons 50

5.1.2 Turbulent Flow 53

5.2 FLOW AROUND AN AIRFOIL 56

5.2.1 Model 56

5.2.2 Laminar Flow 58

5.2.2.1 Numerical Simulation Setup 58

5.2.2.2 Results & Comparisons 59

5.2.3 Turbulent Flow 61

5.3 SECONDARY FLOW IN A CURVED DUCT 66

5.3.1 Numerical Simulation Setup 67

5.3.2 Results & Comparisons 69

5.4 SECONDARY FLOW IN CASCADES 75

5.4.1 2D Cascade 75

5.4.1.1 Numerical Simulation Setup 76

5.4.1.2 Results & Comparisons 77

5.4.2 3D Cascade 81

5.4.2.1 Numerical Simulation Setup 82

5.4.2.2 Results & Comparisons 83

5.5 ROTOR–STATOR INTERACTION 88

5.5.1 ERCOFTAC Radial Pump 88

5.5.1.1 Geometry of the Pump Stage 89

5.5.1.2 Numerical Simulation Setup 90

5.5.1.3 Results & Comparisons 93

5.5.2 Symmetrical Radial Pump 101

5.5.2.1 Geometry of the Pump Stage 101
5.5.2.2 Numerical Simulation Setup 102
5.5.2.3 Results & Comparisons 104

6 CONCLUSIONS 110

7 OUTLOOK 113

BIBLIOGRAPHY 114

List of Figures:

Fig. 2.1 A scale–invariant view of statistical mechanics from cosmic to tachyonic scales [3] 6

Fig. 2.2 Hierarchy of statistical fields for equilibrium eddy–, cluster–, and molecular–dynamic scales and the associated non–equilibrium fields [1] 7

Fig. 2.3 Flow distribution near the source 9

Fig. 2.4 Flow chart of the convective velocity calculation process 14

Fig. 3.1 1D mesh 18

Fig. 3.2 Control volume V [29] 22

Fig. 3.3 Variation of the volume flow rate 24

Fig. 4.1 1D FVM mesh 29

Fig. 4.2 Flow chart of the calculation process of the Navier–Stokes solver in OpenFOAM 34

Fig. 4.3 Flow chart of the calculation process of the Sohrab solver in OpenFOAM 36

Fig. 4.4 Flow chart of the Navier–Stokes solver for dynamic mesh 39

Fig. 4.5 Flow chart of the Sohrab solver for dynamic mesh 41

Fig. 5.1 Laminar boundary layer flow over a flat plate 42

Fig. 5.2 Assumption of the velocity inside the boundary layer 43

Fig. 5.3 Numerical model of the laminar boundary layer flow over a flat plate 45

Fig. 5.4 Grid for the laminar flow over a flat plate 45

Fig. 5.5 Calculated velocity profiles inside the laminar boundary layer 47

Fig. 5.6 Comparison of the velocity profile inside the laminar boundary layer between the numerical and analytical solutions of the modified equation of fluid motion 47

Fig. 5.7 Comparison of the velocity profile inside the laminar boundary layer between the numerical solutions derived from different convective velocities 48

Fig. 5.8 Comparison of the computation time taken by both solvers 49

Fig. 5.9 Comparison of the velocity profile inside the laminar boundary layer between the experiment and numerical solutions 49

Fig. 5.10 Comparison of the magnitude of the convective velocity and boundary layer velocities 52

Fig. 5.11 Comparison of the numerical results influenced by the numerical error 53

Fig. 5.12 Numerical model of the turbulent boundary layer flow over a flat plate 54

Fig. 5.13 Grid for the turbulent flow over a flat plate 54

Fig. 5.14 Comparison of the velocity profile inside the turbulent boundary layer 55

Fig. 5.15 Geometry profile of the airfoil RAF 6 56

Fig. 5.16 Complete grid region for the airfoil at $\alpha = 0°$ 56

Fig. 5.17 Detailed grid for the airfoil at four angles of attack 57

Fig. 5.18 Potential flow over the airfoil 58

Fig. 5.19 Comparison of the drag coefficient for diverse Reynolds numbers 60

Fig. 5.20 Comparison of the drag coefficient for different angles of attack 60

Fig. 5.21 Comparison of the drag coefficient 62

Fig. 5.22 Flow field near the trailing edge calculated by the Navier–Stokes solver 63

Fig. 5.23 Flow field near the trailing edge calculated by the Sohrab solver 63

Fig. 5.24 Comparison of the pressure over the airfoil 64

Fig. 5.25 Comparison of the velocity gradients inside the turbulent boundary layer 64

Fig. 5.26 Geometry of the curved duct 66

Fig. 5.27 Mesh effect on velocity profiles 68

Fig. 5.28 Section of the grid of the curved duct 69

Fig. 5.29 Comparison of the convergence process of the velocity 69

Fig. 5.30 Comparison of computation time taken by both solvers 70

Fig. 5.31 Vector plots of the velocity at various cross sections 71

Fig. 5.32 Sketch of the cross section of the curved duct 72

Fig. 5.33 Comparison of the crosswise velocity profiles inside the bend at θ = 60° 73

Fig. 5.34 Comparison of the streamwise velocity profiles inside the bend at θ = 60° 73

Fig. 5.35 Comparison of the crosswise velocity profiles downstream the bend 74

Fig. 5.36 Comparison of the streamwise velocity profiles downstream the bend 74

Fig. 5.37 3D model of the seven-blade 2D cascade 75

Fig. 5.38 Part grid region of 2D cascade 76

Fig. 5.39 Grids near leading edge and trailing edge 77

Fig. 5.40 Comparison of the convergence processes of the lift coefficient 78

Fig. 5.41 Comparison of the averaged computation time taken by both solvers 78

Fig. 5.42 Comparison of the pressure coefficient 80

Fig. 5.43 Comparison of the lift coefficients 81

Fig. 5.44 3D model of the six-blade annular cascade 81

Fig. 5.45 Part grid of annular cascade 82

Fig. 5.46 Grids near the leading and trailing edges 82

Fig. 5.47 Comparison of the computation time taken by both solvers 83

Fig. 5.48 Comparison of the pressure contours at pressure side of the blade 84

Fig. 5.49 Comparison of the pressure contours at suction side of the blade 85

Fig. 5.50 Position of the polyline for the wake plot 86

Fig. 5.51 Comparison of the wake characteristics 87

Fig. 5.52 ERCOFTAC pump model [55] 89

Fig. 5.53 2D mesh of the complete pump stage 91

Fig. 5.54 Grids near the both leading and trailing edges of the impeller blade 91

Fig. 5.55 Grids near the both leading and trailing edges of the diffuser vane 92

Fig. 5.56 Comparison of computation time taken by both solvers 94

Fig. 5.57 Comparison of the static pressure distribution 95

Fig. 5.58 Comparison of the static pressure distribution in the enlarged blade tip region 96

Fig. 5.59 Comparison of the absolute velocity fields at φ = 5° 97

Fig. 5.60 Comparison of the absolute velocity fields at φ = 25° 97

Fig. 5.61 Comparison of the absolute velocity fields at φ = 35° 98

Fig. 5.62 Comparison of the absolute velocity fields at φ = 40° 98

Fig. 5.63 Velocity triangle at the impeller outlet 98

Fig. 5.64 Circumferential and radial components of the velocities 99

Fig. 5.65 Comparison of relative velocity components in radial gap region at φ = 5° 100

Fig. 5.66 Comparison of relative velocity components in radial gap region at φ = 25° 100

Fig. 5.67 Comparison of relative velocity components in radial gap region at φ = 35° 100

Fig. 5.68 Comparison of relative velocity components in radial gap region at φ = 40° 101

Fig. 5.69 Pump view [54] 101

Fig. 5.70 2D mesh of the complete pump stage 103

Fig. 5.71 Grids near the both leading and trailing edges of the impeller blade . 103

Fig. 5.72 Grids near the both leading and trailing edges of the diffuser vane 104

Fig. 5.73 Comparison of computation time taken by both solvers 105

Fig. 5.74 Comparison of the absolute velocity fields at φ = -10° 106

Fig. 5.75 Comparison of the absolute velocity fields at φ = 0° 107

Fig. 5.76 Comparison of the absolute velocity fields at φ = 10° 108

Fig. 5.77 Comparison of relative velocity components in radial gap region .. 109

Fig. 5.78 Comparison of relative velocity components in radial gap region at φ = 0° 109

Fig. 5.79 Comparison of relative velocity components in radial gap region at φ = 10° 109

List of Tables:

Tab. 5.1 Boundary conditions setup for the laminar boundary layer flow 46

Tab. 5.2 Numerical schemes setup for the laminar boundary layer flow.....46

Tab. 5.3 Boundary conditions of the test cases.....51

Tab. 5.4 Boundary conditions setup for turbulent boundary layer flow 55

Tab. 5.5 Boundary conditions setup for the laminar flow around an airfoil.....59

Tab. 5.6 Numerical schemes setup for the laminar flow around an airfoil.....59

Tab. 5.7 Boundary conditions setup for the turbulent flow around an airfoil...61

Tab. 5.8 Numerical schemes setup for the turbulent flow around an airfoil.....62

Tab. 5.9 Boundary conditions setup for the flow in a curved duct 67

Tab. 5.10 Numerical schemes setup for the flow in a curved duct 67

Tab. 5.11 Boundary conditions setup for the flow in a 2D cascade.....77

Tab. 5.12 Boundary conditions setup for the flow in an annular cascade 83

Tab. 5.13 Specifications of the pump stage [57], [58] 90

Tab. 5.14 Boundary conditions setup for the flow in a radial pump.....92

Tab. 5.15 Numerical schemes setup for the flow in a radial pump.....93

Tab. 5.16 Specifications of the pump stage [54].....102

Tab. 5.17 Boundary conditions setup for the flow in a symmetric radial pump 104

Nomenclature

Variables

A	Area
a, a'	Velocity coefficients
C_d	Drag coefficient
C_l	Lift coefficient
Co	Courant number
C_p	Pressure coefficient
$\mathbf{C}$	Absolute velocity
c	Chord
D^*	Dimensionless distance (distance/boundary layer length)
D	Thermal viscosity
e	Error
$\mathbf{F}_d$	Drag force
$\mathbf{F}_l$	Lift force
$\mathbf{F}$	Force
F	Flux
f_x	Distance rate
f	Distribution function
H	Delivery head, duct height, matrix
Kn	Knudsen number
k	Boltzmann constant, turbulent kinetic energy
l	Span
m	Mass
n	Number, rotating speed
O	Truncation error
Pe	Peclet number
p'	Kinematic pressure
p^*	Dimensionless pressure (p/p_{max})
p	Pressure

Q	Flow rate
R	Radius
Re	Reynolds number
$\mathbf{r}$	Position
r	Distance/axis of rotation
T	Temperature
t	Time
S_ϕ	Source
$\mathbf{S}$	Distance, surfaces
s	Surface
$\mathbf{U}_b$	Moving mesh velocity
$\mathbf{U}$	Circumferential velocity, element velocity, local velocity
$\mathbf{u}$	Atom velocity
u_τ	Friction velocity ($\sqrt{\tau / \rho}$)
V	Volume
$\mathbf{v}_{gh}$, $\mathbf{v}_g$, $\mathbf{v}_h$	Diffusion velocities
v^*	Dimensionless velocity (v/v_{in})
$\mathbf{v}$	Peculiar velocity
$\mathbf{W}$	Relative velocity
$\mathbf{w}$	Convective velocity, system velocity
x, y, z	Coordinates
$y+$	Dimensionless distance between the first grid node and wall
Z	Number

Greek symbols

α	Angle of attack
β	Scale
δ	Difference, distance
ε	Energy dissipation rate
ϕ	Flux, transport variable
Γ	Diffusion
γ	Blending factor
φ	Impeller postion
μ	Viscosity
ν	Kinematic viscosity
ρ	Density
τ	Shear stress
$\boldsymbol{\sigma}$	Stress tensor
$\boldsymbol{\Omega}$	Local velocity / convective velocity
ω	Angular speed
ξ	Similarity variable
$\boldsymbol{\Psi}$	Arbitrary variabel
$\boldsymbol{\psi}$	Invariant function of velocity

Abbreviations

1D	One-dimensional
2D	Two-dimensional
3D	Three-dimensional
ALE	Arbitrary Lagrangian-Eulerian
B. C.	Boundary Condition
BD	Blend Differencing
CD	Central Differencing
CFD	Computational Fluid Dynamics
CV	Control Volume
DGCL	Discrete Geometric Conservation Law
E	East
FVM	Finite-volume Method
GGI	General Grid Interface
HOT	High Order Terms
LCD	Laminar Cluster Dynamics
LED	Laminar Eddy Dynamics
LFD	Laminar Fluid element Dynamics
LHS	Left-hand Side
NVD	Normalized Variable Diagram
N-S	Navier-Stokes
OpenFOAM	Open Field Operation and Manipulation
PS	Pressure Side
RAF	Royal Air Force
RHS	Right-hand Side
SS	Suction Side
SST	Shear Stress Transport
TVD	Total Variation Diminishing
UD	Upwind Differencing
W	West

Superscripts

´	Fluctuating
-	Mean value
n	Time step

Subscripts

1	Impeller inlet
2	Impeller outlet
3	Diffuser inlet
4	Diffuser outlet
β	Scale
a	Axial component
cw	Crosswise
d	Diffuser
E,N,P,W	Cell centre
e,w	Node
free	Freestream
f	Cell face centre
in	Inlet, inside
i	Impeller
J	Scale
M	Modified equation, Sohrab solver
m	Mean value
N-S	Navier-Stokes solver
Num	Numerical
out	Outside
r	Radial component
sw	Streamwise
t	Tangential component, turbulence
u	Circumferential component
w	Wall

1 Introduction

Computational fluid dynamics (CFD) methodology is widely used by engineers and designers in a broad range of industries to gain deeper insight into the effects of fluid dynamics on design processes. The fundamental governing equations for solving most of the CFD problems are the Navier–Stokes equations. Under most conditions, the equations are second order, non–homogenous, non–linear and partial differential equations. The primitive variable formulations of these equations result in the derivation of a large system of non–linear algebraic equations after the finite–volume method (FVM) discretisation is carried out. However, these equations increase the associated computational cost and impede the development of numerical solutions obtained using the Navier–Stokes equations. Hence, it is assumed that engineers would be interested in another equation which does not have non–linear properties.

The scale–invariant theory of statistical mechanics has been developed in recent years. On the basis of a scale–invariant model, Sohrab [1] introduced a linear equation termed the “scale–invariant form of the equation of motion” or the “modified equation of fluid motion”. Preliminary investigations have shown that this modified equation can be extended to solve incompressible flow problems. Several basic flow models were analytically developed using this equation. Consequently, satisfactory correlation between the estimated and experimental data was reached. This result then stimulated further application of this modified equation in the development of CFD codes to obtain more numerical solutions of turbomachinery flows.

1.1 State of the Art of the Modified Equation of Fluid Motion and its Applications

In 1992, a canonical grand unified theory of fields was introduced by Sohrab [2] to give the first impression of the application of the scale–invariant statistical theory of fields to entire spectrum of scales from galactodynamics to subquantum tachyon dynamics. The appearance of this new theory offers another understanding for natural phenomena in diverse branches of modern sciences.

On the basis of the scale–invariant theory of fields, in 1999, Sohrab [3] presented a scale–invariant model of statistical mechanics for equilibrium and non–equilibrium scales. With the help of this model, the invariant definition of the local, convective, and diffusion velocities were introduced. Furthermore the scale–invariant form of the equation of motion which is also called as the modified equation of fluid motion was derived from the scale–invariant form of the Boltzmann equation [1]. In mathematics, this modified equation of fluid motion is nearly identical to the famous Navier–Stokes

equation for incompressible flow. In comparison, the former replaces the local velocity in the convection acceleration component using the convective velocity which is different from the local velocity. Therefore, the non–linear property is eliminated in the modified equation. This important feature uncovers the linear advantage of the modified equation of fluid motion in CFD applications as compared with the Navier–Stokes equation.

In order to prove the ability of the modified equation of fluid motion, several flow models have been analytically developed using this equation. In 1999, Sohrab [4] presented the first analytical solutions of the modified equation of fluid motion for two fundamental flow problems which were the laminar flow over a flat plate and through a circular pipe. For the flat plate problem, the resultant velocity profile inside the boundary layer, expressed by an error function, was compared with the classical theory of Blasius [5]. For the circular pipe, the resultant velocity distribution was compared with the parabolic profile of the classical theory of Hagen and Poiseuille [6]. It was found that both comparisons delivered satisfactory agreement between the modified and classical solutions. In 2006, the modified equation was solved for the laminar flow around a rigid cylinder [7]. Using an ultra low Reynolds number, the analytical solutions for the stream line around the cylinder and the thin boundary layer were compared with the numerical solutions obtained by the Navier–Stokes equations.

The scale–invariant statistical theory also offers another understanding for the turbulence transition. As discussed by Sohrab [8], the potential energy is responsible to preserve the stability of the laminar flow particles. The increase in the particle velocity decreases the potential energy. When the critical velocity is reached, the potential energy vanishes, and the entire laminar field makes the transition to a fully developed turbulence field. This recovers the strong relation between the laminar and turbulence fields for the fluid element, and offers the possibility to solve the modified equation of fluid motion for the turbulent flow scale. An application of this theory can also be found in literatures [8, 9]. The modified equation was solved for the turbulent flow over a flat plate analytically. It is found that both flow fields in the outer laminar layer as well as the inner sub-layer are actually turbulent flow fields that are being convected by a global velocity. However, the associated analytical method was too complex to be directly used in the numerical application. Another treatment for solving the modified equation for turbulence problems is required.

In 2008, for the laminar boundary layer flow problem Sohrab [10] presented a more comprehensive comparison between the predicted velocity profile of the modified equation of fluid motion and several experimental results, including Burgers and Zijnen [11–13], Hansen [14], Dhawan [15], Büttner and Czarske [16] and Nikuradse [17]. Additionally, the predicted velocity profile was compared with the numerical

solution obtained by the Navier–Stokes equations. It was shown that except some of the Nikuradse's data, all experimental data agreed with the prediction of the modified equation however systematically deviate from that of the classical Navier–Stokes equations. These results provide further supports to the CFD application of the modified equation of fluid motion in accuracy.

1.2 Motivations

All the investigations above contribute the understanding of the modified equation of fluid motion in mathematics, and give the proof for the applications of the modified equation with some fundamental flow models analytically. However these analytical methods cannot go further to solve the modified equation for complicated boundary conditions. This limits the scientific investigations of the modified equation in CFD applications.

In order to expose the potential advantage of the modified equation of fluid motion for turbomachinery applications, the present research goes deeper insight into the modified equation using the numerical method of approach. In this context, the open source CFD toolbox OpenFOAM (Open Field Operation and Manipulation) is employed to establish a numerical solver using the modified equation. This developed solver must support the required mesh topologies and be available to use the FVM to discretize the modified equation of fluid motion. One key attribute of this solver is that with all supported flow models, it must produce accurate solution with fast and reliable convergence. Furthermore, as compared to the similar CFD solvers dealing with the Navier–Stokes equations, it must deliver more efficient solutions for the same CFD problems, in terms of its linear property.

1.3 Present Contributions

In this work, the investigation is restricted to incompressible flow to avoid dealing with more technical conditions on the scale–invariant form of the energy equation.

Referring to the motivations described above, this research makes the following contributions to the numerical solution of the modified equation of the fluid motion:

(1) On the basis of the work done by Sohrab, the derivation of the modified equation of the fluid motion for incompressible flow is presented with more emphasis on its linear property as compared to the Navier–Stokes equation for incompressible flow.

(2) In order to apply the finite difference method to discretize the modified equation, a detailed analysis of the present discretisation technique for the modified equation is performed. As compared with the Navier–Stokes equation, the numerical errors resulted from the discretisation of the modified equation, including the truncation and discretisation errors are discussed as well as the stability conditions.

(3) On the basis of the analysis above, a novel numerical solver is established by using the open source FVM code OpenFOAM. A corresponding iteration technology is determined for the solver to dedicate reliable and efficient solutions to the algebraic linear equations derived from the FVM discretisation of the modified equation. In addition, a dynamic mesh solver developed on the basis of the arbitrary Lagrangian Eulerian (ALE) method is also built up to resolve the transient flow problems.

(4) Finally, the developed solver is applied to solve the modified equation for several flow models including the fundamental boundary layer problems, flow around an airfoil, turbulent secondary flow in a curved duct and cascades, and rotor–stator interactions in radial pumps. As compared with the Navier–Stokes solver, the mesh resolution, numerical scheme and computation time required for the developed solver are investigated. In order to examine the accuracy of the solver, the obtained results are validated by comparing them with the former experimental and analytical data reported in the literatures.

1.4 Thesis Structure

The thesis is organized as follows. In Chapter 2, the scale–invariant statistical theory related to the fluid dynamics scales is briefly introduced and a detailed description of the modified equation of fluid motion for incompressible flow is presented. In Chapter 3, the numerical errors caused by the space discretisation of the modified equation of fluid motion are discussed to give an access for the solver establishing by using the FVM. In Chapter 4, a developed solver for the modified equation of fluid motion is introduced in the OpenFOAM environment. In Chapter 5, the simulations of several flow models are discussed to explain the applications of the developed solver. In Chapter 6, the present research is concluded. In Chapter 7, suggestions for future investigations are proposed.

2 Modified Equation of Fluid Motion

In 1999, Sohrab [3] gave an introduction to the physical and mathematical foundation of a scale–invariant statistical theory of fields. The construction of the associated mathematical models would be greatly helpful to some fundamental conceptual difficulties which are encountered in engineering applications.

In this chapter, on the basis of the scale–invariant theory, the derivation of the modified equation of fluid motion will be discussed in detail. Furthermore, the linear advantage of this equation will be emphasized, as compared with the classical Navier–Stokes equations. In addition, the time averaged expression of the modified equation for turbulent flow will also be delivered.

2.1 Scale–invariant Model of Statistical Mechanics

A scale–invariant model was established for both equilibrium statistical mechanics, which is governed by the Gibbs function, and non–equilibrium field, which is based on the Liouville, Boltzmann and Maxwell methods. As shown in Fig. 2.1, the broad scales from the exceedingly large scale of cosmic to the minute scale of quantum optics are included in the scale–invariant model [3]. For equilibrium dynamics, the scales are equilibrium galacto–, planetary–, hydrosystem–, fluid–element–, eddy–, cluster–, molecular–, atomic–, subatomic–, chromo–, and tachyon–dynamics respectively corresponding to the scale β = *g, p, h, f, e, c, m, a, s, k* and *t*. As well, the associated non–equilibrium scales are listed on the right–hand side (RHS) in Fig. 2.1. The classical fluid mechanics covers the fields LFD, LED and LCD which are short for Laminar Fluid element Dynamics, Laminar Eddy Dynamics and Laminar Cluster Dynamics.

In Fig. 2.2 four scales are selected to declare the relationship between the neighboring scales. As discussed in [3], each field described by a distribution function:

$$f_\beta(\mathbf{r}_\beta, t_\beta, \mathbf{u}_\beta) \tag{2.1}$$

defines a “system”, which is composed of an ensemble of “element”, as well, each element is composed of an ensemble of small particles viewed as point mass “atoms”. For example, in Fig. 2.2 the element velocity of the smaller scale *J* becomes the atom velocity of the larger scale *J+1*:

$$\mathbf{U}_J = \mathbf{u}_{J+1} \tag{2.2}$$

and the system velocity in scale J becomes the element velocity in scale $J+1$:

$$\mathbf{w}_J = \mathbf{U}_{J+1} \tag{2.3}$$

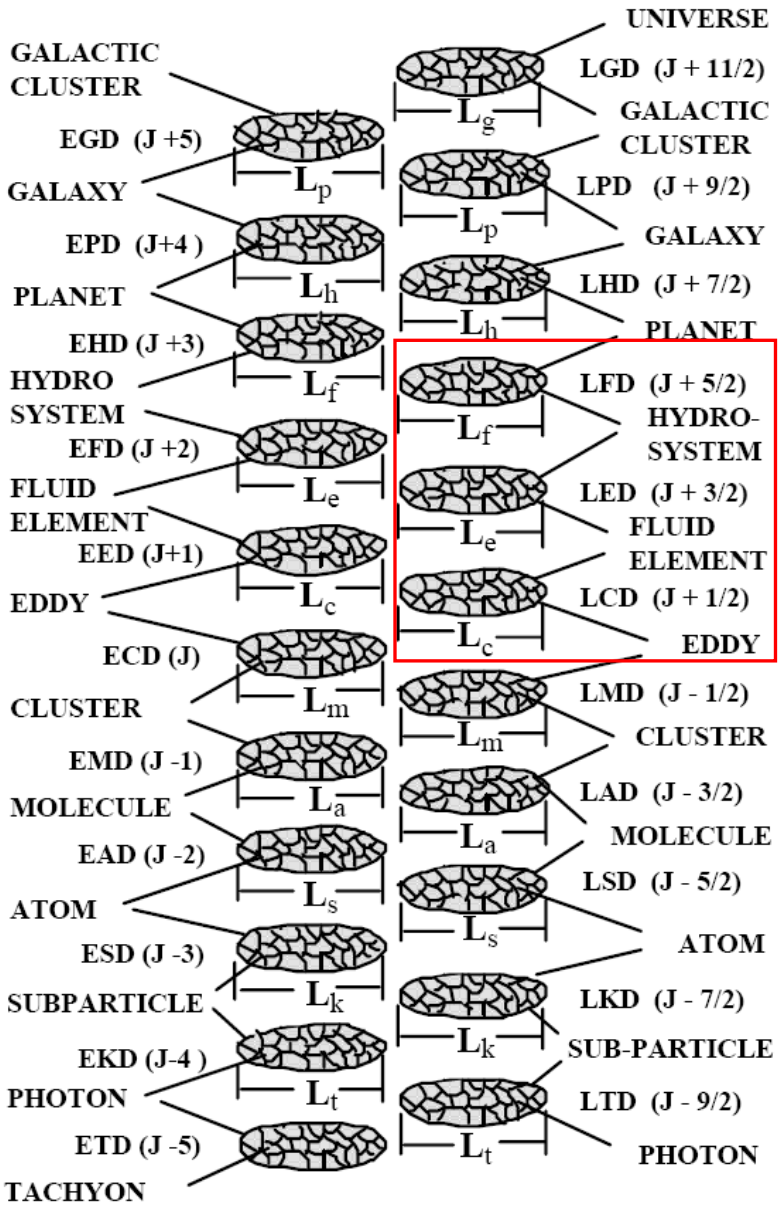


Fig. 2.1 A scale–invariant view of statistical mechanics from cosmic to tachyonic scales [3]

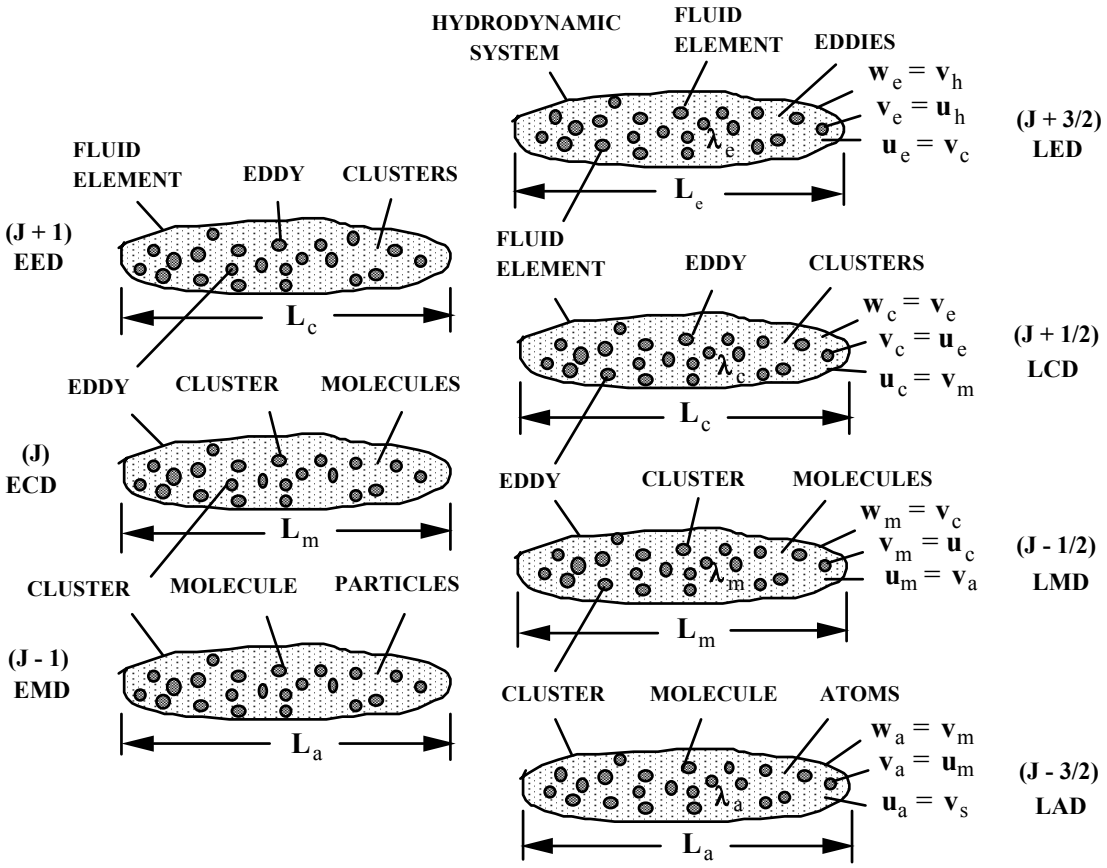


Fig. 2.2 Hierarchy of statistical fields for equilibrium eddy–, cluster–, and molecular–dynamic scales and the associated non–equilibrium fields [1]

According to the scale–invariant theory, flow parameters can be defined in a scale–invariant way. As introduced in [3], the invariant definition of the density ρ_β, and the relation between the velocities of element **U** (or local velocity), atom **u** and system **w** at neighboring scales are given as follows:

$$\rho_\beta = n_\beta m_\beta = m_\beta \int f_\beta d\mathbf{u}_\beta \tag{2.4}$$

$$\mathbf{u}_\beta = \mathbf{U}_{\beta\text{-}1} \tag{2.5}$$

$$\mathbf{U}_\beta = \rho_\beta^{-1} m_\beta \int \mathbf{u}_\beta f_\beta d\mathbf{u}_\beta \tag{2.6}$$

$$\mathbf{w}_\beta = \mathbf{U}_{\beta+1} \tag{2.7}$$

2.2 Convective, Diffusion and Local Velocities

At scale β, the system velocity $\mathbf{w}_\beta$ can also be understood as mean velocity or convective velocity [18]:

$$\mathbf{w}_\beta = \overline{\mathbf{U}_\beta} \tag{2.8}$$

Following the classical theory [19–21], the fluid particle transfers inside a physical system due to two processes: diffusion and convection which is also known as advection. Therefore, using the scale–invariant form, at the scale β, the local velocity $\mathbf{U}_\beta$ can be expressed in terms of the convective velocity $\mathbf{w}_\beta$ and the diffusion velocity $\mathbf{v}_{\beta gh}$:

$$\mathbf{U}_\beta = \mathbf{w}_\beta + \mathbf{v}_{\beta gh} \tag{2.9}$$

For most flow problems, the fluid diffusion contains two components:

$$\mathbf{v}_{\beta gh} = \mathbf{v}_{\beta g} + \mathbf{v}_{\beta h} \tag{2.10}$$

In detail,

$$\mathbf{v}_{\beta g} = -D_\beta \nabla \ln(\rho_\beta) \tag{2.11}$$

which is related to the density dependent diffusion, and

$$\mathbf{v}_{\beta h} = -\nu_\beta \nabla \ln(\mathbf{U}_\beta) \tag{2.12}$$

which stands for the viscous influence to the diffusion.

Incompressible flow assumption gives

$$\nabla \rho_\beta = 0 \tag{2.13}$$

Hence, in the scope of the present work, only the viscous component affects the diffusion:

$$\mathbf{v}_{\beta gh} = \mathbf{v}_{\beta h} = -\nu_\beta \nabla \ln(\mathbf{U}_\beta) \tag{2.14}$$

Substituting Eq. (2.14) into Eq. (2.9), one obtains

$$\mathbf{U}_\beta = \mathbf{w}_\beta + \mathbf{v}_{\beta h} \tag{2.15}$$

Eq. (2.15) plays a significant role in the derivation of the modified equation of fluid motion. In order to deliver a better understanding of this equation, a flow distribution arising from the source *S* is illustrated in Fig. 2.3. As shown in this figure, when

$$\mathbf{U}_\beta = \mathbf{v}_{\beta h} \tag{2.16}$$

the fluid only performs the diffusion motion, the green region will be the final result of the fluid distribution. On the other hand, when

$$\mathbf{U}_\beta = \mathbf{w}_\beta \tag{2.17}$$

the fluid only performs the convection motion, the resultant fluid distribution will be the yellow stream. However as mentioned above, in most cases the fluid particles transport with both diffusion and convection motions. In Fig. 2.3, this combination is the field inside the red ellipse.

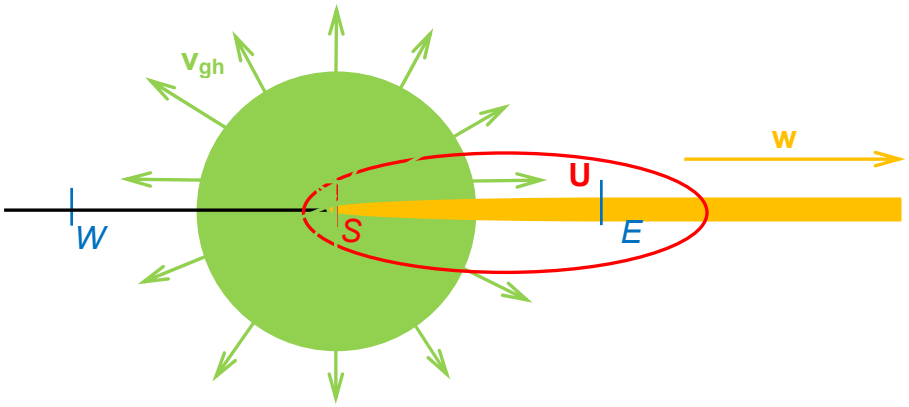


Fig. 2.3 Flow distribution near the source

2.3 Modified Equation of Fluid Motion for Incompressible Flow

The modified equation of fluid motion is derived from the scale–invariant form of the Boltzmann Equation [1]. The Boltzmann equation, often known as the Boltzmann transport equation can be used to study how a fluid transports the momentum, thus deriving transport properties, such as viscosity [22].

In this section, the derivation process of the modified equation of fluid motion will be described in detail. Furthermore, this modified equation will be compared with the Navier–Stokes equation to highlight its potential advantages.

2.3.1 Derivation of the Modified Equation of Fluid Motion

On the basis of the theory of non–equilibrium statistical mechanics, Sohrab [1] gave the scale–invariant form of the Boltzmann equation in the absence of the body force as follows:

$$\frac{\partial f_\beta}{\partial t} + \mathbf{u}_\beta \cdot \nabla f_\beta = \frac{\delta f_\beta}{\partial t} \tag{2.18}$$

Multiplying Avogadro–Loschmidt number with an arbitrary invariant function of velocity $\boldsymbol{\psi}_\beta$ and integrating them to both sides of Eq. (2.18), one obtains the Enskog equation of change in the scale–invariant form:

$$\frac{\partial}{\partial t}(n_\beta \overline{\boldsymbol{\psi}_\beta}) + \nabla \cdot (n_\beta \overline{\boldsymbol{\psi}_\beta \mathbf{u}_\beta}) = \int \boldsymbol{\psi}_\beta \cdot \frac{\delta f_\beta}{\delta t} d\mathbf{u}_\beta \tag{2.19}$$

Defining

$$\boldsymbol{\psi}_\beta = m_\beta \mathbf{u}_\beta \tag{2.20}$$

in Eq. (2.19), one has

$$\frac{\partial}{\partial t}(n_\beta m_\beta \mathbf{U}_\beta) + \nabla \cdot (n_\beta m_\beta \mathbf{U}_\beta \mathbf{U}_\beta) = \int m_\beta \mathbf{u}_\beta \cdot \frac{\delta f_\beta}{\delta t} d\mathbf{u}_\beta \tag{2.21}$$

Note that the density definition in the statistical mechanics is

$$\rho_\beta = n_\beta m_\beta \tag{2.22}$$

Eq. (2.21) can be rewritten in the form:

$$\frac{\partial}{\partial t}(\rho_\beta \mathbf{U}_\beta) + \nabla \cdot (\rho_\beta \mathbf{U}_\beta \mathbf{U}_\beta) = \int m_\beta \mathbf{U}_\beta \cdot \frac{\delta f_\beta}{\delta t} d\mathbf{U}_\beta \tag{2.23}$$

According to Sohrab [1], under the Stokes assumption, the term on the RHS of Eq. (2.23) results in

$$\frac{\partial}{\partial t}(\rho_\beta \mathbf{U}_\beta) + \nabla \cdot (\rho_\beta \mathbf{U}_\beta \mathbf{U}_\beta) = -\nabla p_\beta + \frac{1}{3}\mu \nabla (\nabla \cdot \mathbf{U}_\beta) \tag{2.24}$$

which shows the thermodynamic and hydrodynamic pressure. For incompressible flow, the hydrodynamic pressure can be neglected, therefore Eq. (2.23) is simplified as follows:

$$\rho_\beta \cdot \frac{\partial}{\partial t}\mathbf{U}_\beta + \rho_\beta \nabla \cdot (\mathbf{U}_\beta \mathbf{U}_\beta) = -\nabla p_\beta \tag{2.25}$$

Referring to Eq. (2.15) , the second term in Eq. (2.25) can be expanded in the form:

$$\begin{aligned} \rho_\beta \nabla \cdot (\mathbf{U}_\beta \mathbf{U}_\beta) &= \rho_\beta \nabla \cdot [(\mathbf{w}_\beta + \mathbf{v}_{\beta h})\mathbf{U}_\beta] \\ &= \rho_\beta \nabla \cdot (\mathbf{w}_\beta \mathbf{U}_\beta) + \rho_\beta \nabla \cdot (\mathbf{v}_{\beta h} \mathbf{U}_\beta) \end{aligned} \tag{2.26}$$

Since the convective velocity is not a variable for the local flow field, Eq. (2.26) can be transferred as follows:

$$\rho_\beta \nabla \cdot (\mathbf{U}_\beta \mathbf{U}_\beta) = \rho_\beta \mathbf{w}_\beta \cdot \nabla \mathbf{U}_\beta + \rho_\beta \nabla \cdot (\mathbf{v}_{\beta h} \mathbf{U}_\beta) \tag{2.27}$$

Substituting Eq. (2.12) into Eq. (2.27), the second term on the RHS of Eq. (2.27) becomes

$$\begin{aligned} \rho_\beta \nabla \cdot (\mathbf{v}_{\beta h} \mathbf{U}_\beta) &= \rho_\beta \nabla \cdot \{[-\nu_\beta \nabla \ln(\mathbf{U}_\beta)]\mathbf{U}_\beta\} \\ &= -\rho_\beta \nu_\beta \nabla \cdot [(\frac{1}{\mathbf{U}_\beta} \cdot \nabla \mathbf{U}_\beta)\mathbf{U}_\beta] \\ &= -\rho_\beta \nu_\beta \nabla^2 \mathbf{U}_\beta \end{aligned} \tag{2.28}$$

Consequently, Eq. (2.27) becomes

$$\rho_\beta \nabla \cdot (\mathbf{U}_\beta \mathbf{U}_\beta) = \rho_\beta \mathbf{w}_\beta \cdot \nabla \mathbf{U}_\beta - \rho_\beta \nu_\beta \nabla^2 \mathbf{U}_\beta \tag{2.29}$$

As a result, Eq. (2.23) can be expressed as follows:

$$\frac{\partial}{\partial t}\mathbf{U}_\beta + \mathbf{w}_\beta \cdot \nabla \mathbf{U}_\beta - \nu_\beta \nabla^2 \mathbf{U}_\beta = -\frac{\nabla p_\beta}{\rho_\beta} \tag{2.30}$$

which is known as the modified equation of fluid motion for incompressible flow. On the other hand, for compressible flow, this modified equation owns the similar expression. The detailed discussion can be found in the literature [1].

As presented in Eq. (2.30), the modified equation of fluid motion is a second partial differential equation. It is emphasized that the convective velocity $\mathbf{w}_\beta$ is presented in the convection term of the equation. This introduced parameter can be calculated by Eq. (2.7) or Eq. (2.8). In other words, on one hand, for the current flow scale it is the system velocity as presented in Eq. (2.7), being equal to the local velocity at the next larger scale; on the other hand, the convective velocity can be understood as the mean velocity for the current scale system as shown in Eq. (2.8), which can be constructed theoretically for the simple flow model, e.g. the laminar boundary layer flow over a flat plate. In order to clarify these properties, numerical methods to determine the convective velocity will be discussed in Chapter 5.

The relation between the convective velocities at two neighboring scales, the laminar flow scale and the potential flow scale is under consideration. For potential flow scale, the viscosity vanishes, so the diffusion velocity becomes

$$\mathbf{v}_{\beta h} = -\nu_\beta \nabla \ln(\mathbf{U}_\beta) = 0 \tag{2.31}$$

Continuing, $\mathbf{U}_\beta$ gives

$$\mathbf{U}_\beta = \mathbf{w}_\beta + \mathbf{v}_{\beta h} = \mathbf{w}_\beta \tag{2.32}$$

Therefore, for potential flow $\mathbf{w}_\beta$ in Eq. (2.30) is replaced by $\mathbf{U}_\beta$ and the scale–invariant form of the Euler equation is obtained as follows:

$$\frac{\partial}{\partial t}\mathbf{U}_\beta + \mathbf{U}_\beta \cdot \nabla \mathbf{U}_\beta = -\frac{\nabla p_\beta}{\rho_\beta} \tag{2.33}$$

This result indicates that, at the potential flow scale, the modified equation of fluid motion produces the same solution as the Euler equation. Therefore, according to Eq. (2.7) the solution for the convective velocity at the laminar flow scale can be constructed by using the velocity at the next larger scale, the potential flow scale. This relation can be expressed by the following equation:

$$\mathbf{w}_{laminar} = \mathbf{U}_{potential} \tag{2.34}$$

For simplicity, the subscript symbol β will be neglected for the scale–invariant form equations in the following discussions.

2.3.2 Relation between the Modified Equation and the Navier–Stokes Equation

The classical Navier–Stokes equation for incompressible flow arises from applying Newton's second law to fluid motion, together with the assumption that the fluid stress is the sum of a diffusing viscous term, plus a pressure term [23]. For incompressible flow, the Navier–Stokes equation can be expressed as:

$$\frac{\partial}{\partial t}\mathbf{U} + \mathbf{U} \cdot \nabla \mathbf{U} - \nu \nabla^2 \mathbf{U} = -\frac{\nabla p}{\rho} \tag{2.35}$$

with the consideration of the continuity equation

$$\nabla \cdot \mathbf{U} = 0 \tag{2.36}$$

For small Knudsen number, $Kn << 1$, the derivation of the Navier–Stokes equation for incompressible flow from the Boltzmann equation is also possible [24]. This recovers the connection between the Navier–Stokes equation and the modified equation of fluid motion, that both of them can be applied to analyze transport phenomena.

Restricting attention to the second term of Eq. (2.30) and Eq. (2.35) on the left–hand side (LHS), it is obvious that the convective velocity **w** can replace the local velocity **U**. In mathematics, this important feature eliminates the non–linearity. As a numerical investigation, the present research is mainly focused on the linear property of the modified equation of fluid motion.

2.4 Reynolds Averaged Modified Equation of Fluid Motion

In CFD, it is very difficult for the numerical method to account the complete unsteady turbulence problems because of the complexity and significant computer time requirement. In most cases, the time averaged equation can deliver the net effect of the turbulence perturbation.

Similar to the Reynolds–averaged Navier–Stokes equation (RANS), the time averaged form of the modified equation of fluid motion in conjunction with the isotropic turbulence viscosity hypothesis is considered for the further numerical applications to handle the turbulence problems.

Turbulent eddies produce the fluctuation of velocity and pressure in both space and time. Given the mean velocity, $\mathbf{U}_m$, mean pressure p_m and the corresponding turbulent fluctuation values, $\mathbf{U}'$ and p', the transient expressions for velocity and pressure at the turbulence scale are

$$\mathbf{U}_t = \mathbf{U}_{tm} + \mathbf{U}'_t \tag{2.37}$$

$$p_t = p_{tm} + p'_t \tag{2.38}$$

Arising from the quantum mechanics, Sohrab [25] described another understanding of the turbulence transition and presented that the convective velocity at the turbulence scale can be derived from the local velocity at the laminar scale as:

$$\mathbf{w}_{tm} = \mathbf{U}_{laminar} \tag{2.39}$$

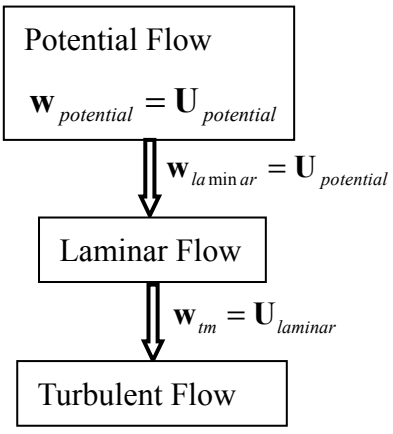


Fig. 2.4 Flow chart of the convective velocity calculation process

Therefore, concluding Eqs. (2.32), (2.34) and (2.39), the convective velocity for the turbulence scale is sourced from the potential flow scale, and developed in the laminar flow scale, as shown in the flow chart of Fig. 2.4.

Following this turbulence transition analysis, the transient expression for the convective velocity **w** is

$$\mathbf{w}_t = \mathbf{w}_{tm} + \mathbf{w}_t' \tag{2.40}$$

According to the averaging method, the averaged value for the transient pressure and velocities are

$$\overline{p_t} = p_{tm} \quad \overline{p_t'} = 0 \tag{2.41}$$

$$\overline{\mathbf{U}_t} = \mathbf{U}_{tm} \quad \overline{\mathbf{U}'_t} = 0 \tag{2.42}$$

$$\overline{\mathbf{w}_t} = \mathbf{w}_{tm} \quad \overline{\mathbf{w}_t'} = 0 \tag{2.43}$$

Applying the results above, at the turbulence scale, the time averaged results for each component in Eq. (2.30) can be obtained as follows:

$$\overline{\frac{\partial \mathbf{U}_t}{\partial t}} = \frac{\partial \mathbf{U}_{tm}}{\partial t} \tag{2.44}$$

$$\begin{aligned} \overline{\mathbf{w}_t \cdot \nabla \mathbf{U}_t} &= \nabla \overline{(\mathbf{w}_t \mathbf{U}_t)} \\ &= \nabla (\mathbf{w}_{tm} \mathbf{U}_{tm} + \overline{\mathbf{w}_t' \mathbf{U}_t'}) \\ &= \nabla (\mathbf{w}_{tm} \mathbf{U}_{tm}) + \nabla (\overline{\mathbf{w}_t' \mathbf{U}_t'}) \end{aligned} \tag{2.45}$$

$$\overline{\nabla^2 (\mathbf{U}_t)} = \nabla^2 \mathbf{U}_{tm} \tag{2.46}$$

$$\overline{\frac{\nabla (p_t)}{\rho_t}} = \frac{\nabla p_{tm}}{\rho_t} \tag{2.47}$$

In isotropic turbulence, the convective velocity shows the transient property of the system. Its transient fluctuation component should own the equivalent averaged value as that of the local velocity. In other words, one has

$$\overline{\mathbf{w}_t'\mathbf{U}_t'} = \overline{\mathbf{U}_t'\mathbf{U}_t'} \tag{2.48}$$

Consequently, Eq. (2.45) becomes

$$\overline{\mathbf{w}_t \cdot \nabla \mathbf{U}_t} = \nabla(\mathbf{w}_{tm}\mathbf{U}_{tm}) + \nabla(\overline{\mathbf{U}_t'\mathbf{U}_t'}) \tag{2.49}$$

Finally, the Reynolds–averaged modified equation of fluid motion for turbulent flow is

$$\frac{\partial \mathbf{U}_{tm}}{\partial t} + \mathbf{w}_{tm} \cdot \nabla \mathbf{U}_{tm} - \nu \cdot \nabla^2 \mathbf{U}_{tm} = -\frac{\nabla p_{tm}}{\rho_{tm}} - \nabla(\overline{\mathbf{U}_t'\mathbf{U}_t'}) \tag{2.50}$$

Eq. (2.50) represents the Reynolds–averaged modified equation of fluid motion. In this equation, all time dependent terms are dropped since only steady states are considered. For incompressible flow, the Reynolds stress $\rho(\mathbf{U}_t'\mathbf{U}_t')$ has the same structure and dimension as the viscous stress tensor. However, instead of stress, it is just a re–worked contribution of the fluctuating velocities to the change of the averaged ones. Same as the RANS, the Reynolds stress in the Reynolds–averaged modified equation of fluid motion can be described by the turbulent kinetic energy (k) and the rate of energy dissipation (ε). This feature indicates that, the Reynolds stress in Eq. (2.50) can be handled by the turbulence model in numerical applications. Especially considering the bottleneck of the analytical method described in Chapter 1, the present analysis offers another opportunity for the modified equation of fluid motion to get the access to solve the turbulence problems. In addition, this will also simplify the coding of the numerical solver for modeling the turbulent flow.

3 Numerical Error

In order to solve the modified equation of fluid motion for turbomachinery flow problems, a numerical solver is required. FVM is well suited for the numerical simulation of the Navier–Stokes equation in the field fluid dynamics. Considering the relation between the modified and Navier–Stokes equations, in this research, the FVM is used to establish the numerical solver for the modified equation.

FVM is a discretisation method for representing and evaluating partial differential equations in the form of algebraic equations [26–27]. Inevitably, the discretisation process generates truncation and discretisation errors in the numerical solution. Before establishing the numerical solver, an estimation of these numerical errors derived from the discretisation of the modified equation of fluid motion is presented in this chapter.

Both modified and Navier–Stokes equations describe the transport phenomena of fluid dynamics. By starting from the generic transport equation, this chapter discusses the truncation and discretisation errors of the aforementioned two equations. Then the emphasis is placed on the differences of the numerical errors between these two equations to highlight the linear advantages of the modified equation in numerical applications.

The transport equation for incompressible flow can be written in the differential form:

$$\frac{\partial}{\partial t}\phi + \mathbf{\Omega}\cdot\nabla\phi - \nu\nabla^2\phi = -\frac{S_\phi}{\rho} \tag{3.1}$$

where ϕ is the transport variable, ν is the diffusion coefficient, $\mathbf{\Omega}$ stands for the local velocity **U** in the Navier–Stokes equation, or the convective velocity **w** in the modified equation of fluid motion and S_ϕ is the source that can either create or destroy ϕ.

On the LHS of Eq. (3.1), the third component, which describes the diffusion property, contains the second derivative of ϕ in space, therefore Eq. (3.1) is a second–order equation. As discussed by Patankar [28], the order of the discretisation has to be equal or higher than the order of Eq. (3.1) to deliver a satisfactory accuracy for the discretisation solution.

3.1 Taylor Series Expansion

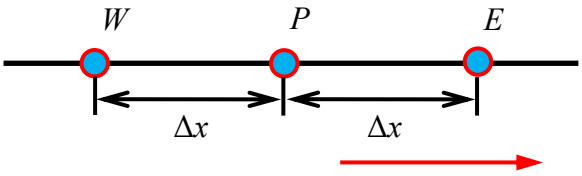


Fig. 3.1 1D mesh

Assuming the one–dimensional (1D) grids with the uniform space interval Δx, as sketched in Fig. 3.1, the Taylor series expansion for the transport variable ϕ on this grid will be

$$\phi_E = \phi_P + \frac{\partial \phi}{\partial x}\Delta x + \frac{\partial^2 \phi}{\partial x^2}\frac{\Delta x^2}{2!} + \frac{\partial^3 \phi}{\partial x^3}\frac{\Delta x^3}{3!} + \frac{\partial^4 \phi}{\partial x^4}\frac{\Delta x^4}{4!} + HOT \tag{3.2}$$

$$\phi_W = \phi_P - \frac{\partial \phi}{\partial x}\Delta x + \frac{\partial^2 \phi}{\partial x^2}\frac{\Delta x^2}{2!} - \frac{\partial^3 \phi}{\partial x^3}\frac{\Delta x^3}{3!} + \frac{\partial^4 \phi}{\partial x^4}\frac{\Delta x^4}{4!} + HOT \tag{3.3}$$

where *HOT* is short for high order terms.

Substracting Eq. (3.3) from Eq. (3.2) produces

$$\phi_E - \phi_W = 2\frac{\partial \phi}{\partial x}\Delta x + 2\frac{\partial^3 \phi}{\partial x^3}\frac{\Delta x^3}{3!} + HOT \tag{3.4}$$

Dividing by $2\Delta x$ on both sides of Eq. (3.4), ϕ can be expressed as follows:

$$\frac{\partial \phi}{\partial x} = \frac{\phi_E - \phi_W}{2\Delta x} - \frac{\partial^3 \phi}{\partial x^3}\frac{\Delta x^2}{3!} + HOT \tag{3.5}$$

Setting $O(\Delta x^p)$ as the truncation error in Taylor series, one obtains

$$\nabla \phi = \frac{\partial \phi}{\partial x} = \frac{\phi_E - \phi_W}{2\Delta x} + O(\Delta x^2) \tag{3.6}$$

where

$$O(\Delta x^2) = -\frac{\partial^3 \phi}{\partial x^3}\frac{\Delta x^2}{3!} + HOT \tag{3.7}$$

Now $\nabla \phi$ is expressed by a second order accurate central difference interpolation.

If Δx is replaced by a time step size Δt, and points W, P, E are replaced by time steps $(n\text{-}1)^{th}$, n^{th} and $(n+1)^{th}$, the second order accurate temporal derivative can be obtained as:

$$\frac{\partial \phi}{\partial t} = \frac{\phi^{n+1} - \phi^{n-1}}{2\Delta t} + O(\Delta t^2) \tag{3.8}$$

If the first order accurate discretisation solution (Eq. (3.2)) is applied to the temporal derivation, at point P, one has

$$\frac{\partial \phi_P}{\partial t} = \frac{{\phi_P}^{n+1} - {\phi_P}^{n}}{\Delta t} + O(\Delta t) \tag{3.9}$$

which is well known as the forward method.

On the other hand, $\nabla^2\phi$ can be obtained by the summation of Eq. (3.2) and Eq. (3.3), as follows:

$$\phi_E + \phi_W = 2\phi_P + 2\frac{\partial^2 \phi}{\partial x^2}\frac{\Delta x^2}{2!} + 2\frac{\partial^4 \phi}{\partial x^4}\frac{\Delta x^4}{4!} + HOT \tag{3.10}$$

Dividing by Δx^2 on both sides of Eq. (3.10), ϕ can be explicitly expressed:

$$\frac{\partial^2 \phi}{\partial x^2} = \frac{\phi_W - 2\phi_P + \phi_E}{\Delta x^2} - 2\frac{\partial^4 \phi}{\partial x^4}\frac{\Delta x^2}{4!} + HOT \tag{3.11}$$

Setting

$$O(\Delta x^2) = -2\frac{\partial^4 \phi}{\partial x^4}\frac{\Delta x^2}{4!} + HOT \tag{3.12}$$

the second order accurate discretisation for $\nabla^2\phi$ can be written in the form:

$$\nabla^2 \phi = \frac{\phi_W - 2\phi_P + \phi_E}{\Delta x^2} + O(\Delta x^2) \tag{3.13}$$

Referring to Eqs. (3.6), (3.9) and (3.13), finally the discretisation solution of the transport equation on the 1D mesh can be obtained as follows:

$$\frac{{\phi_P}^{n+1} - {\phi_P}^{n}}{\Delta t} + \mathbf{\Omega} \cdot \frac{{\phi_E}^{n} - {\phi_W}^{n}}{2\Delta x} - \nu \frac{{\phi_W}^{n} - 2{\phi_P}^{n} + {\phi_E}^{n}}{\Delta x^2} = -\frac{S_\phi}{\rho} \tag{3.14}$$

where the time discretisation is first order and the space discretisation is second order. Since the source will not make influence to the discretisation difference between the modified and Navier–Stokes equations, this component is not discussed in the present work.

3.2 Truncation Error

Referring to Eqs. (3.2), (3.7) and (3.12) discussed in section 3.1, the truncation error of the transport equation (Eq. (3.1)) is

$$\frac{\partial^2\phi}{\partial t^2}\frac{\Delta t}{2} + HOT + \mathbf{\Omega}\cdot(\frac{\partial^3\phi}{\partial x^3}\frac{\Delta x^2}{3!} + HOT) + \nu(2\frac{\partial^4\phi}{\partial x^4}\frac{\Delta x^2}{4!} + HOT)$$

Since the time step Δt can be well controlled during the numerical simulation, and for most fluid the kinematic viscosity v is much smaller compared to the velocity $\mathbf{\Omega}$, the convection component in the transport equation produces the maximum truncation error. On the other hand, the Navier–Stokes equation discretisation result differs from the modified equation of fluid motion only in the convection component. Consequently, it becomes more valuable to concentrate the analysis on this component.

For Navier–Stokes equation, one has

$$\mathbf{\Omega}\cdot(\frac{\partial^3\phi}{\partial x^3}\frac{\Delta x^2}{3!} + HOT) = \mathbf{U}\cdot(\frac{\partial^3\phi}{\partial x^3}\frac{\Delta x^2}{3!} + HOT) \tag{3.15}$$

while, for the modified equation of fluid motion, one has

$$\mathbf{\Omega}\cdot(\frac{\partial^3\phi}{\partial x^3}\frac{\Delta x^2}{3!} + HOT) = \mathbf{w}\cdot(\frac{\partial^3\phi}{\partial x^3}\frac{\Delta x^2}{3!} + HOT) \tag{3.16}$$

For both equations, the special interval Δx can be trended to zero, so that the truncation error trends to zero. However, in practice, one always has

$$\Delta x > 0$$

Hence, the truncation error is strongly dependent on $\mathbf{\Omega}$.

During the numerical computation of the Navier–Stokes equation, the local velocity $\mathbf{U}$, experiences large oscillation and converges in the end of the computation. Consequently, the associated truncation error will be strongly influenced by this numerical oscillation. However, in the modified equation of fluid motion, the

convective velocity is determined from another scale and preserved as a vector field with fixed values for the current scale, thus avoiding the aforementioned oscillation during the numerical computation process. Therefore, after the discretisation is carried out, although the modified equation cannot reduce the truncation error, independent of the accuracy order, it can produce a relative stable truncation error as compared to the Navier–Stokes equation. Further, this advantage will make more influence to the accuracy of the numerical solution when the first order accurate discretisation method is applied, e. g. the upwind method. A numerical example about this will be discussed in section 5.1.

On the other hand, this interesting result stimulates the comparison of the stability between these two equations. According to [28], the stability condition for the transport equation Eq. (3.1) is

$$\frac{\mathbf{\Omega}}{\nu / \Delta x} \leq 2 \tag{3.17}$$

which is also well known as the Peclet number.

Referring to $Pe \leq 2$, for Navier–Stokes equation, one has

$$\frac{\mathbf{U}}{\nu / \Delta x} \leq 2 \tag{3.18}$$

Since $\mathbf{U}$ is a variable varying during the numerical solution, one can only reduce Δx to preserve the stability of the equation. Hence a high quality mesh is required for the Navier–Stokes equation discretisation to increase the stability of the associated numerical solution.

While for the modified equation of fluid motion, one has

$$\frac{\mathbf{w}}{\nu / \Delta x} \leq 2 \tag{3.19}$$

The convective velocity $\mathbf{w}$ has been determined for the whole computational domain. With the help of the fixed convective velocity, Δx is relaxed from the requirement of the stability of the discretisation. As compared to the Navier–Stokes equation, a relative coarse mesh can also be accepted by the modified equation of fluid motion for the numerical discretisation. In order to support this analysis, numerical examples concerning the mesh issues will be discussed in sections 5.1 and 5.3 for two–dimensional (2D) and three–dimensional (3D) flow problems respectively.

The truncation error has been theoretically discussed in this section from the Taylor series expansion. It has been noted that, since the truncation error cannot be eliminated in the numerical approximation, a more stable error leads to a faster computation convergence by reducing the numerical oscillation. This analysis reveals the linear advantage of the modified equation of fluid motion when the second order accurate discretisation method is applied to the equations. However, in practice, most FVM software employs a less accurate method to resolve the convection component, thus introducing the so–termed numerical diffusion to the solution. In the following section, this practical numerical error will be discussed in detail.

3.3 Discretisation Error – Numerical Diffusion

The discretisation of the transport equation is under consideration by using FVM.

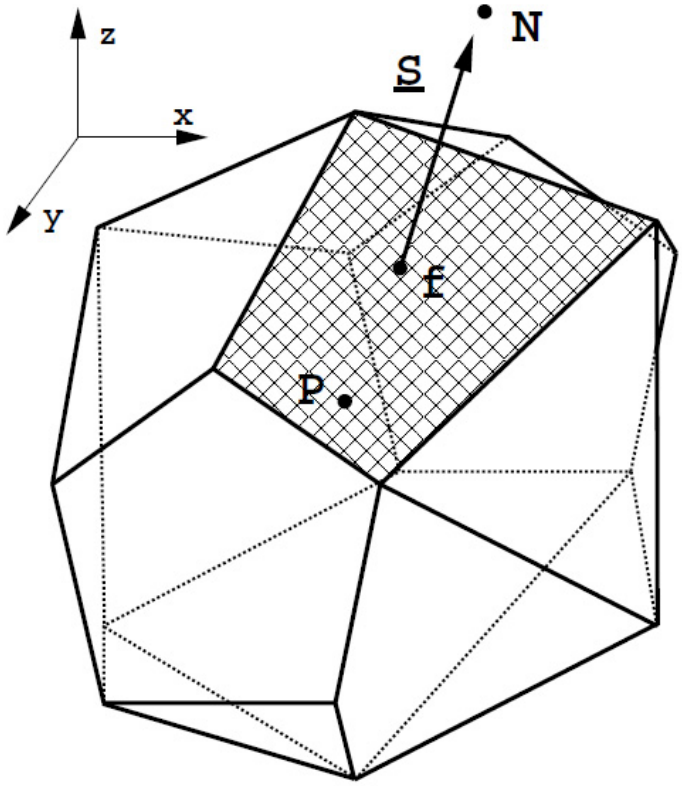


Fig. 3.2 Control volume V [29]

The space discretisation for FVM requires a domain termed control volume (CV). As sketched in Fig. 3.2, a computational node P lies at the centre of the CV of the volume V while N stands for the centres of the neighbor control volumes. For simplicity, all faces of the CV will be marked with f, which also represents the point in the middle of the face, as the face area vector $\mathbf{S}_f$ points outwards from the P CV [29]. Thus, the discretisation of the transport equation on this CV becomes

$$\frac{d}{dt}\int_V \phi dV + \oint_S ds\cdot(\mathbf{\Omega})\phi - \oint_S ds\cdot\mathbf{q}_\phi = -\int_V S_\phi(\phi)dV \tag{3.20}$$

where $\mathbf{q}_\phi$ stands for the diffusion component.

Then, the emphasis is placed on the discretisation of the convection components of the two equations to further highlight the linear advantages of the modified equation in FVM applications.

Taking into account $\nabla\mathbf{\Psi}$ over the CV in Fig. 3.2, the discretised form of the Gauss' theorem can be expressed as follows:

$$(\nabla\cdot\mathbf{\Psi})V_P = \sum_f s\cdot\mathbf{\Psi}_f \tag{3.21}$$

Using Eq. (3.21), the discretisation of the convection term is obtained:

$$\begin{aligned}\int_{V_P}\nabla\cdot(\mathbf{\Omega}\phi)dV &= \sum_f \mathbf{S}\cdot(\mathbf{\Omega}\phi)_f \\ &= \sum_f \mathbf{S}\cdot\mathbf{\Omega}_f\phi_f \\ &= \sum_f F\cdot\phi_f\end{aligned} \tag{3.22}$$

where F in Eq. (3.22) represents the flux through the cell faces:

$$F = \mathbf{S}\cdot\mathbf{\Omega}_f \tag{3.23}$$

Then, three traditional types of discretisation schemes are considered to estimate the numerical diffusions of the modified and Navier–Stokes equations. These three schemes are:

1) Central Differencing (CD),

2) Upwind Differencing (UD) and

3) Blend Differencing (BD)

Since the other high order discretisation schemes are just the sum of CD and the numerical diffusions, such as Total Variation Diminishing (TVD) and Normalized Variable Diagram (NVD), they are not discussed in the present work.

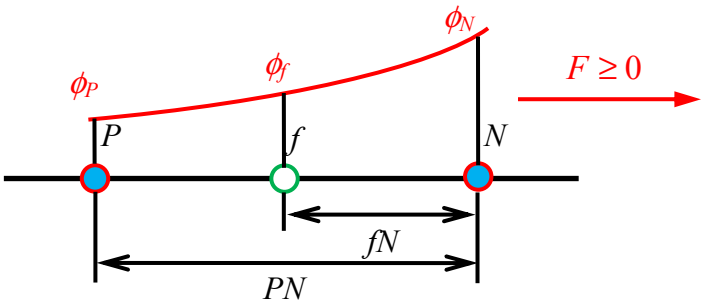


Fig. 3.3 Variation of the transport variable ϕ

The variation of ϕ between the nodes P and N is sketched in Fig. 3.3. According to the truncation error analysis in Section 3.2, a second order accurate scheme is required to satisfy the discretisation of the convection component (Eq. (3.22)). Hence, applying the CD method, one has

$$\phi_f = f_x \phi_P + (1 - f_x)\phi_N \tag{3.24}$$

where f_x is defined as the ratio of distances fN and PN:

$$f_x = \frac{fN}{PN} \tag{3.25}$$

However, the method CD is an unbounded differencing scheme which can not represent the flow direction in the discretisation process. Usually, when strong convection phenomenon appears in the computation, CD will introduce another numerical oscillation to the solution, thus violating the boundedness of the solution [28, 30]. In section 4.1.1, this numerical oscillation will be revealed by an equation discretisation on a 1D mesh.

The boundedness problem can be solved by the UD method which uses differencing biased in the direction determined by the sign of the characteristic speeds. Applying scheme UD to the convection case in Fig. 3.3, one obtains

$$\phi_f = \begin{cases} \phi_f = \phi_P & \text{when} \quad F \geq 0 \\ \phi_f = \phi_N & \text{when} \quad F < 0 \end{cases} \tag{3.26}$$

Obviously, UD is a first order accurate method which ensures the boundedness at a heavy expense at the accuracy of the solution. The numerical diffusion introduced by this method will be discussed later.

Comparing the two discretisation methods described above, it is found that CD offers the high accuracy but violates the boundedness of the solution, while UD guarantees the boundedness but violates the order of accuracy. Between accuracy and boundedness, an alternative discretisation method attempts to solve the problem. Ferziger and Peric [31] combined the methods CD and UD and gave another discretisation method termed BD:

$$\phi_f = \gamma(\phi_f)_{UD} + (1-\gamma)(\phi_f)_{CD} \tag{3.27}$$

or

$$\phi_f = \left[\gamma \max(\mathrm{sgn}(F), 0) + (1-\gamma) f_x\right]\phi_P + \left[\gamma \min(\mathrm{sgn}(F), 0) + (1-\gamma)(1-f_x)\right]\phi_N \tag{3.28}$$

In simplification, let $F > 0$, and Eq. (3.28) becomes

$$\phi_f = \left[\gamma + (1-\gamma) f_x\right]\phi_P + \left[(1-\gamma)(1-f_x)\right]\phi_N \tag{3.29}$$

where γ is a blending factor, $0 \le \gamma \le 1$, which determines how much numerical diffusion is introduced to the solution.

The discretisation error or numerical diffusion introduced by BD can be quantitatively determined by the difference between the product of CD and BD, since CD represents a second order accurate solution. It is found that, for the convection component, this numerical diffusion will be further influenced by the variable **Ω** which represents the key difference between the modified and Navier–Stokes equations. Therefore, in the following discussions the linear advantage of the fixed convective velocity in the modified equation of fluid motion will be highlighted.

The difference in ϕ_f between Eq. (3.24) and Eq. (3.29) is

$$\begin{aligned}
\delta\phi_f &= (\phi_f)_{CD} - (\phi_f)_{BD} \\
&= \left[f_x\phi_P + (1-f_x)\phi_N\right] - \left\{\left[\gamma + (1-\gamma) f_x\right]\phi_P + \left[(1-\gamma)(1-f_x)\right]\phi_N\right\} \\
&= \left[f_x\gamma - \gamma\right]\phi_P + \left[(1-f_x)\gamma\right]\phi_N \\
&= (1-f_x)\gamma(\phi_N - \phi_P)
\end{aligned} \tag{3.30}$$

The variation of ϕ_f can be computed by ϕ_N and ϕ_P as follows:

$$\nabla \phi_f = \frac{(\phi_N - \phi_P)}{PN} \tag{3.31}$$

Substituting Eq. (3.31) into Eq. (3.30), one obtains

$$\delta\phi_f = (1 - f_x)\gamma \cdot PN \cdot \nabla \phi_f \tag{3.32}$$

Then, referring to Eq. (3.22), for the convection component, one has

$$\begin{aligned} e = \sum_f F \cdot \delta\phi_f &= \sum_f \mathbf{S} \cdot \mathbf{\Omega}_f \cdot (1 - f_x)\gamma \cdot PN \cdot \nabla \phi_f \\ &= \sum_f \mathbf{S} \cdot \left[(1 - f_x)\gamma \cdot \mathbf{\Omega}_f \cdot PN \right] \cdot \nabla \phi_f \end{aligned} \tag{3.33}$$

Considering the kinematic viscosity calculated from the Reynolds number at *PN*:

$$\Gamma = \mathbf{\Omega}_f \cdot PN / Re \tag{3.34}$$

a numerical kinematic viscosity can be defined as follows:

$$\Gamma_{Num} = (1 - f_x)\gamma \cdot \mathbf{\Omega}_f \cdot PN \tag{3.35}$$

Substituting Eq. (3.35) into Eq. (3.33), the discretisation error of the convection component can be clarified as a combination of numerical kinematic viscosity and the gradient of ϕ:

$$e = \sum_f \mathbf{S} \cdot (\Gamma_{Num} \cdot \nabla \phi)_f \tag{3.36}$$

Since Eq. (3.36) is similar to the discretisation result of the diffusion component in the transport equation, it is always termed the numerical diffusion.

The property of the numerical diffusion is addressed by Eq. (3.35) and Eq.(3.36). It can be found that this term is linearly dependent on the variable $\mathbf{\Omega}$, grid spacing *PN* and the gradient of ϕ. Consequently, a high quality mesh can reduce the error while a large convection velocity enlarges the error. For a uniform flow,

$$\nabla \phi = 0 \tag{3.37}$$

The numerical diffusion vanishes. However, this condition is scarce in turbomachinery problems. Therefore, the numerical diffusion cannot be neglected in turbomachinery applications. In fact, this term is the key difference between the modified and Navier–Stokes equations in error analysis.

Consider this term in the detailed equations. For the Navier–Stokes equation, one has

$$\Gamma_{Num\_N\text{-}S} = (1 - f_x)\gamma \cdot \mathbf{U}_f \cdot PN \tag{3.38}$$

while for the modified equation of fluid motion, one obtains

$$\Gamma_{Num\_M} = (1 - f_x)\gamma \cdot \mathbf{w}_f \cdot PN \tag{3.39}$$

It is clear that the discretisations of the two equations produce different numerical errors by the numerical diffusion, thus depending on the difference between the local velocity and the convective velocity of the flow field in the computational domain. However, similar to the truncation error result, a fixed convective velocity in the modified equation preserves a more stable numerical diffusion as compared to the Navier–Stokes equation. In comparison, a more stable truncation error leads to a faster computation convergence, while a more stable numerical diffusion not only further accelerates the convergence, but also results in a more accurate solution in numerical computations. These features will strongly affect the numerical solutions of the two equations in CFD applications. In Chapter 5, these issues will be discussed by detailed numerical simulations.

4 Developed Solver

During the past years, Sohrab [1–4, 7–10] solved the modified equation of fluid motion for several flow models analytically. However a unique analytical solution is only feasible for idealized and simple flow boundary conditions. For engineering applications, it is required to implement the numerical method which allows not only better representation of the complex flow phenomena, but also permits more accurate modeling of the boundaries.

The stream of numerical solution technique for the Navier–Stokes equation is FVM. It is originally developed as a special finite difference formulation. For the present research, it also stimulates the approximation solution to the modified equation of fluid motion since the modified equation owns the similar mathematic expression as compared with the Navier–Stokes equation.

A general FVM code for the modified equation of fluid motion is difficult to write because of the problem of specifying domains other than common geometrical figures. OpenFOAM is a C++ toolbox for the development of customized numerical solvers for continuum mechanics problems, including CFD [32]. This open source software offers a satisfactory structure for the FVM discretisation. One distinguishing feature of OpenFOAM is its syntax for partial differential equations that closely resembles the equations being solved. This syntax makes it possible to create a numerical solver for the modified equation of fluid motion with relative ease. In this research, several versions of OpenFOAM, including OpenFOAM 1.5, 1.5–dev, 1.6, 1.7 and 1.6–ext, were tested to establish a suitable numerical solver for turbomachinery applications. In the end, the specialized version OpenFOAM 1.5–dev which offers general grid interface (GGI) function is selected to develop the FVM solver for the modified equation of fluid motion.

In Chapter 3, the numerical errors derived from the space discretisation of the modified equation of fluid motion have been analytically estimated by comparing them with those obtained by the Navier–Stokes equation. The linear advantage of the fixed convective velocity in the stability of the numerical solution has been highlighted. In this chapter, the modified equation is completely discretised on a 1D mesh to investigate its linear property on the resultant algebraic equations. Furthermore, this linear property gives help to establish an efficient numerical solver in the OpenFOAM environment.

4.1 1D Discretisation

A 1D case is taken as an example to show the discretisation process of the modified equation of fluid motion.

For 1D problem, in a steady state, Eq. (2.30) results in:

$$w\frac{dU}{dx}-\nu\frac{d^2U}{dx^2}=-\frac{dp'}{dx} \tag{4.1}$$

where

$$p'=\frac{p}{\rho} \tag{4.2}$$

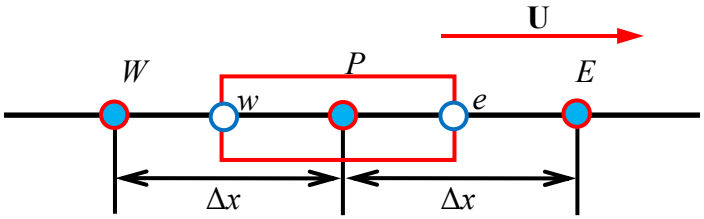


Fig. 4.1 1D FVM mesh

A uniform 1D mesh is shown in Fig. 4.1. Due to the velocity–pressure coupling effect, w and e are treated as nodes while W, P and E are assumed as the cell centres. As compared with Fig. 3.2, w, e stand for f, and W, E represent N. Noting the space discretisation suggestion in [28], integrating Eq. (4.1) on the mesh, one has

$$\begin{aligned}&[(wAU)_e-(wAU)_w]-\nu[(A\frac{dU}{dx})_e-(A\frac{dU}{dx})_w]\\&=-[(Ap')_e-(Ap')_w]\end{aligned} \tag{4.3}$$

Assuming a constant area A for all cell faces and transferring the velocity coefficients with the pre–defined values at the cell centres give

$$[(wU)_e-(wU)_w]-\nu[(\frac{U_E-U_P}{\Delta x}-\frac{U_P-U_W}{\Delta x}]=p'_w-p'_e \tag{4.4}$$

As discussed in Chapter 3, most discretisation schemes are based on two schemes CD and UD. Therefore, for simplicity the discretisation results of Eq. (4.4) are presented only by using these two aforementioned schemes.

4.1.1 CD Discretisation

First, taking the scheme CD to do the discretisation for $(wU)_e$ and $(wU)_w$, one obtains

$$(wU)_e = \frac{w_P + w_E}{2} \cdot \frac{U_P + U_E}{2} \tag{4.5}$$

$$(wU)_w = \frac{w_W + w_P}{2} \cdot \frac{U_W + U_P}{2} \tag{4.6}$$

Then, substituting Eq. (4.5) and Eq. (4.6) into Eq. (4.4) gives

$$\begin{aligned}&(2\nu \frac{1}{\Delta x} + \frac{w_E - w_W}{4})U_P - (\nu \frac{1}{\Delta x} + \frac{w_W + w_P}{4})U_W - (\nu \frac{1}{\Delta x} - \frac{w_P + w_E}{4})U_E \\ &= p'_w - p'_e\end{aligned} \tag{4.7}$$

It is obvious that the velocity coefficients contain the convective velocity w instead of the local velocity U and the resultant algebraic equation is a linear one.

It should be noted that CD scheme produces the velocity coefficient for U_E as:

$$a_E = \nu \frac{1}{\Delta x} - \frac{w_P + w_E}{4} \tag{4.8}$$

When a large convection velocity is presented, a_E becomes negative, $a_E < 0$. This result explains the unbounded property of CD scheme described in section 3.3.

4.1.2 UD Discretisation

Then, the scheme UD is considered for the discretisation process. Setting $U_w > 0$, $U_e > 0$, one has

$$(wU)_e = w_P \cdot U_P \tag{4.9}$$

$$(wU)_w = w_W \cdot U_W \tag{4.10}$$

Then, substituting Eqs. (4.9) and (4.10) into Eq. (4.4) gives

$$(2\nu\frac{1}{\Delta x}+w_P)U_P-(\nu\frac{1}{\Delta x}+w_W)U_W-(\nu\frac{1}{\Delta x})U_E=p'_w-p'_e \tag{4.11}$$

This linear algebraic expression is similar to that obtained from the CD scheme, where the velocity coefficient contains the convective velocity *w* instead of the local velocity *U*. In addition, the velocity coefficient for U_E becomes

$$a_E=\nu\frac{1}{\Delta x} \tag{4.12}$$

which is independent of the convection velocity. This result indicates the bounded property of UD scheme described in section 3.3.

For simplicity, the discretisation results from the UD scheme are taken for further discussions.

Let

$$a_P=2\nu\frac{1}{\Delta x}+w_P \tag{4.13}$$

$$a_W=\nu\frac{1}{\Delta x}+w_W \tag{4.14}$$

Finally, the discretisation of the modified equation of fluid motion on 1D mesh gives

$$a_P\cdot U_P-a_W\cdot U_W-a_E\cdot U_E=p'_e-p'_w \tag{4.15}$$

On the other hand, carrying out the same discretisation for the Navier–Stokes equation produces

$$a'_P=2\nu\frac{1}{\Delta x}+U_P \tag{4.16}$$

$$a'_W=\nu\frac{1}{\Delta x}+U_W \tag{4.17}$$

$$a'_E=\nu\frac{1}{\Delta x} \tag{4.18}$$

$$a'_P \cdot U_P - a'_W \cdot U_W - a'_E \cdot U_E = p'_e - p'_w \tag{4.19}$$

Therefore, it is clear that for the Navier–Stokes equation, the velocity coefficients have to be calculated from the local velocity at the cell centre. This result leads to a non–linear algebraic equation, and indicates the non–linear properties of the Navier–Stokes equation.

4.2 Solver Algorithm & Iteration

The open–source CFD toolbox OpenFOAM, is employed to implement the numerical investigation of the modified equation of fluid motion. Using the FVM structure in this software, a new numerical solver based on the modified equation of fluid motion is programmed. The developed solver is named as Sohrab solver. In order to point out the difference between the iterations of both Sohrab and Navier–Stokes solvers, this section presents a detailed comparison between the velocity and pressure calculations executed by both solvers.

4.2.1 Navier–Stokes Solver

In a steady state, the Navier–Stokes equation (Eq. (2.35)) can be simplified as:

$$\mathbf{U} \cdot \nabla \mathbf{U} - \nu \nabla^2 \mathbf{U} = -\frac{\nabla p}{\rho} \tag{4.20}$$

After the FVM discretisation is carried out, referring to Fig. 3.2, the LHS of the Eq. (4.20) is expressed as follows:

$$\mathbf{U} \cdot \nabla \mathbf{U} - \nu \nabla^2 \mathbf{U} = a'_P \mathbf{U}_P + \sum_N a'_N \mathbf{U}_N \tag{4.21}$$

where $\mathbf{U}_N$ stands for the velocities at the neighboring points (e. g. points *W*, *E* in section 4.1) of the point *P*.

As described by Jasak [29], in OpenFOAM, an important matrix $H(\mathbf{U})$ is introduced as:

$$H(\mathbf{U}) = -\sum_N a'_N \mathbf{U}_N \tag{4.22}$$

Then, the momentum equation can be constructed as:

$$a'_P \mathbf{U}_P = H(\mathbf{U}) - \nabla p \tag{4.23}$$

Expressing **U** in the explicit method gives

$$\mathbf{U}_P = \frac{H(\mathbf{U})}{a'_P} - \frac{\nabla p}{a'_P} \tag{4.24}$$

Integrating the continuity equation on the cell face gives

$$\nabla \cdot \mathbf{U} = \sum_f A \cdot \mathbf{U}_f = 0 \tag{4.25}$$

Substituting Eq. (4.24) into Eq. (4.25), the pressure equation can be obtained as follows:

$$\nabla \cdot (\frac{1}{a'_P} \nabla p) = \sum_f A \cdot (\frac{H(\mathbf{U})}{a'_P})_f \tag{4.26}$$

Eq. (4.26) helps the solver compute the pressure (or pressure gradient) with pressure boundary conditions. Then, the velocity $\mathbf{U}_P$ can be initially calculated by Eq. (4.24) and updated by iterations until a convergence is reached.

A simple flow chart can represent this calculation process in OpenFOAM. As shown in Fig. 4.2, the matrix $H(\mathbf{U})$ will be updated in each velocity iteration ($\mathbf{U}_P$ iteration) until the velocity and pressure reach their convergence in the current time step. Using these converged values, the code recalculates the coefficients of a'_P and a'_N to initialize the next time step. When the residual of the velocity $\mathbf{U}_P$ reaches the computation convergence criteria, the whole calculation converges, and the difference of a'_P and a'_N between the neighboring time steps can be neglected. In this method, the non–linear properties of the Navier–Stokes equation are solved by the iteration technology as well as the velocity–pressure coupling.

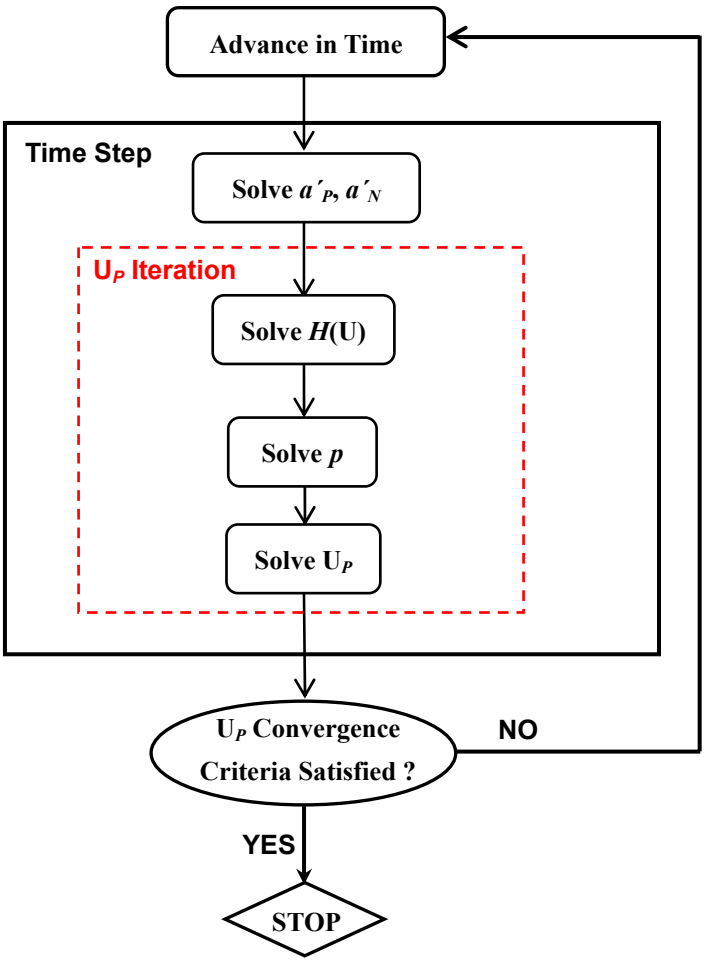


Fig. 4.2 Flow chart of the calculation process of the Navier–Stokes solver in OpenFOAM

4.2.2 Sohrab Solver

The discretisation of the modified equation of fluid motion delivers another expressions of a_P and a_N as shown in Eqs. (4.12) – (4.14). Noting that these two coefficients are composed of the convective velocity, and independent of the local velocity which is the variable to be solved, it can be found that a_P and a_N can be directly fixed at the initial step of the FVM computation. Hence, these two coefficients are not necessary to join the time step iterations in the solver. This feature points out another advantage of the linear property of the modified equation in numerical applications. However, the velocity–pressure coupling difficulty exists in the equation, therefore the iteration method is still required when coding the Sohrab solver. Certainly, the iteration of the Sohrab solver is different to that of the Navier–Stokes solver described in Fig. 4.2. The flow chart in Fig. 4.3 shows the corresponding new iteration process of the Sohrab solver.

Comparing the FVM computation processes of the two solvers, it is found that the Sohrab solver saves the re–calculation process of the coefficients a_P and a_N in each time step. This feature of the Sohrab solver reduces the total running time of the FVM computation from the viewpoint of the programming of the code. This analysis will be proved by the numerical examples presented in Chapter 5.

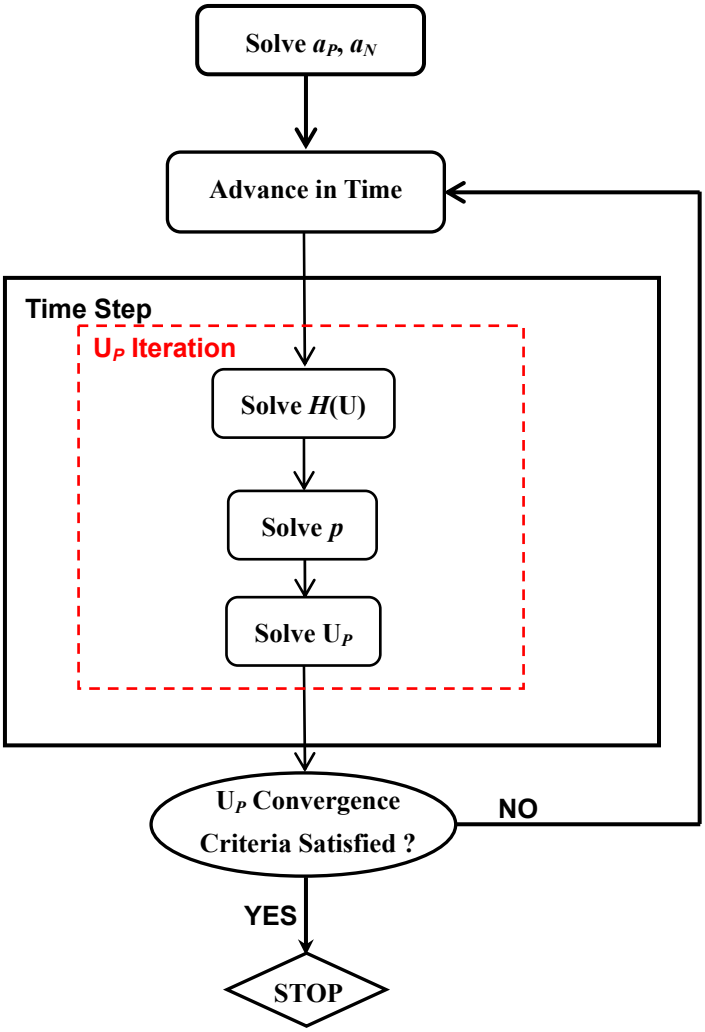


Fig. 4.3 Flow chart of the calculation process of the Sohrab solver in OpenFOAM

4.3 Dynamic Mesh Solver

In the sections before, the Sohrab solver has been established for the steady state flow. On the basis of this analysis, the dynamic mesh solver is further discussed in this section.

In the transient flow application, e.g. rotor–stator interaction, the physical domain of rotor rotates in time. Since this moving domain experiences large rotation, it becomes necessary to solve the flow field in the moving grids which is always referred to as dynamic mesh. A popular formulation for solving flow problems on dynamic meshes is the closely arbitrary Lagrangian Eulerian solution (ALE) [33], [34]. In these considerations, the numerical algorithm designed for computing the flow problems on dynamic mesh typically incurs the computation of some geometric quantities involving the grid position and mesh velocity [35]. These quantities and the time–integrating fluxes on dynamic mesh have to be covered by the Discrete Geometric Conservation Law (DGCL) where an additional term related to the "mesh velocity" $\mathbf{U}_b$ will be presented. This conservation law states that the computation of the geometric parameters must be performed in such a way that, independently of the mesh motion, the resultant numerical algorithm preserves the state of a uniform flow [36]. In OpenFOAM, the moving mesh results in the change of cell volume V and mesh motion fluxes ϕ expressed by the following equations [37]:

$$\frac{d}{dt}\int_V \phi dV + \oint_S ds\cdot(\boldsymbol{\Omega}-\mathbf{U}_b)\phi - \oint_S ds\cdot\mathbf{q}_\phi = -\int_V s(\phi)dV \tag{4.27}$$

$$\frac{d}{dt}\int_V dV - \oint_S ds\cdot\mathbf{U}_b = 0 \tag{4.28}$$

$$\oint_S d\mathbf{S}\cdot\mathbf{U}_b = \sum_f \int_{S_f} d\mathbf{S}\cdot\mathbf{U}_b = F_m \tag{4.29}$$

There has been sufficient numerical evidence showing that satisfying the DGCL considerably improves the time accuracy of numerical computation on dynamic meshes [38], [39].

4.3.1 Navier–Stokes Solver

In OpenFOAM, the dynamic mesh solver for the Navier–Stokes equation is established based on Eq. (4.27) rather than Eq. (3.20). The mesh motion flux appears in all convection terms and needs to be accounted for algorithmically. In the developed version of OpenFOAM 1.5–dev, the relative flux, instead of the absolute flux, is applied

to solve the equation, hence the dynamic mesh solver is not yet consistently integrated with static mesh solver, e. g. the solver analyzed in Fig. 4.2.

The dynamic mesh solver for the Navier–Stokes equation can be principally described in Fig. 4.4. In order to satisfy the DGCL, mesh update and flux correction are added to the initialization of the solver. Furthermore, the velocity coefficients a'_P and a'_N have to be updated in the current time step according to the new flux. Referring to Eqs. (4.16) – (4.18), the updated velocity coefficients become

$$a'_P = 2\nu \frac{1}{\Delta x} + (U_P - U_{bP}) \tag{4.30}$$

$$a'_W = \nu \frac{1}{\Delta x} + (U_W - U_{bW}) \tag{4.31}$$

$$a'_E = \nu \frac{1}{\Delta x} \tag{4.32}$$

On the other hand, all the velocity terms are computed with the relative flux rather than the absolute one, due of Eq. (4.27). This issue is highlighted in the time step iterations and PISO loop in Fig. 4.4. Therefore, the output of the solver is also influenced by Eq. (4.27), where the resultant velocity is absolute to the mesh motion, while the flux is relative to the motion. This result explains that why the first step in the solver is to make the flux absolute to the mesh motion. The present analysis for the dynamic mesh Navier–Stokes solver offers experience for the coding of the associated Sohrab solver.

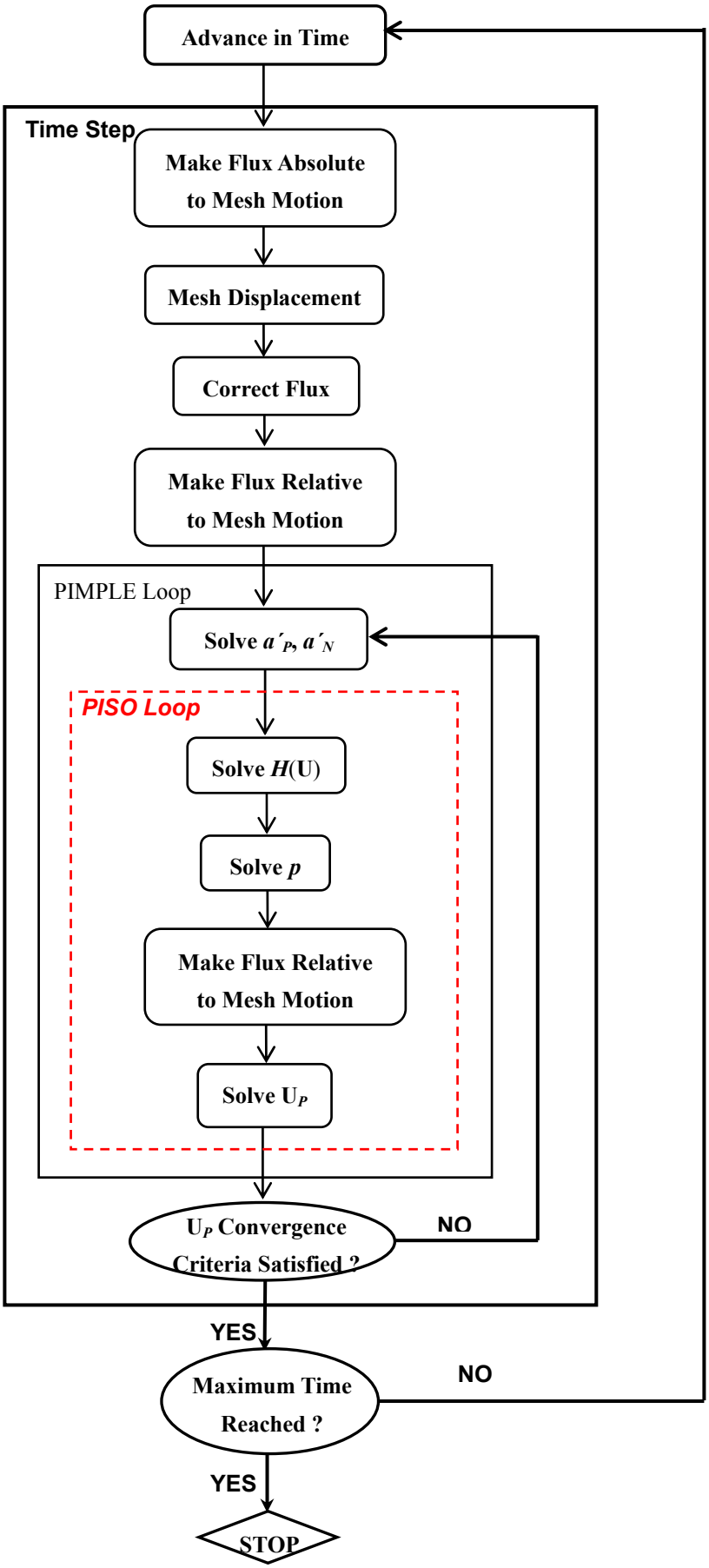


Fig. 4.4 Flow chart of the Navier–Stokes solver for dynamic mesh

4.3.2 Sohrab Solver

DGCL plays the same role in the geometry conservation of the numerical applications of the modified equation of fluid motion for dynamic mesh problems. The relative flux is helpful to clarify the efficiency of the Sohrab solver in solving the dynamic mesh problems. The modified equation consequently becomes

$$\frac{d}{dt}\int_V \phi dV + \oint_S ds\cdot(\mathbf{w}-\mathbf{U}_b)\phi - \oint_S ds\cdot\mathbf{q}_\phi = -\int_V s(\phi)dV \tag{4.33}$$

Eq. (4.33) and the analysis for the dynamic mesh solver of the Navier–Stokes indicate that the required velocity coefficients a_P and a_N for the dynamic mesh solver cannot be preserved as fixed parameters for all the time steps. Instead, they have to be updated in each time step as expressed in the following equations:

$$a_P = 2\nu\frac{1}{\Delta x} + (w_P - U_{bP}) \tag{4.34}$$

$$a_W = \nu\frac{1}{\Delta x} + (w_W - U_{bW}) \tag{4.35}$$

$$a_E = \nu\frac{1}{\Delta x} \tag{4.36}$$

This process points out the difference between the static mesh and the dynamic mesh simulations when applying the Sohrab solver.

Comparing the Sohrab solver to the Navier–Stokes solver, it is found that the additional convective velocity calculation process is naturally different to the Navier–Stokes solver. As explained in section 4.3.1, a'_P and a'_N in the Navier–Stokes solver would be corrected several times in each time step, because of the numerical iteration and physical mesh moving process. However in the Sohrab solver, the corresponding items are updated only due to the later reason and require no iteration for further corrections. Consequently the dynamic mesh solver for the modified equation of fluid motion is generally somewhat more costly than the steady state solver, but still saves time by comparing it to the Navier–Stokes solver. In order to declare this feature, the structure of the Sohrab solver for moving mesh is described by a flow chart in Fig. 4.5. Furthermore, numerical examples for consuming time comparisons of the dynamic mesh solvers will be presented in Chapter 5.

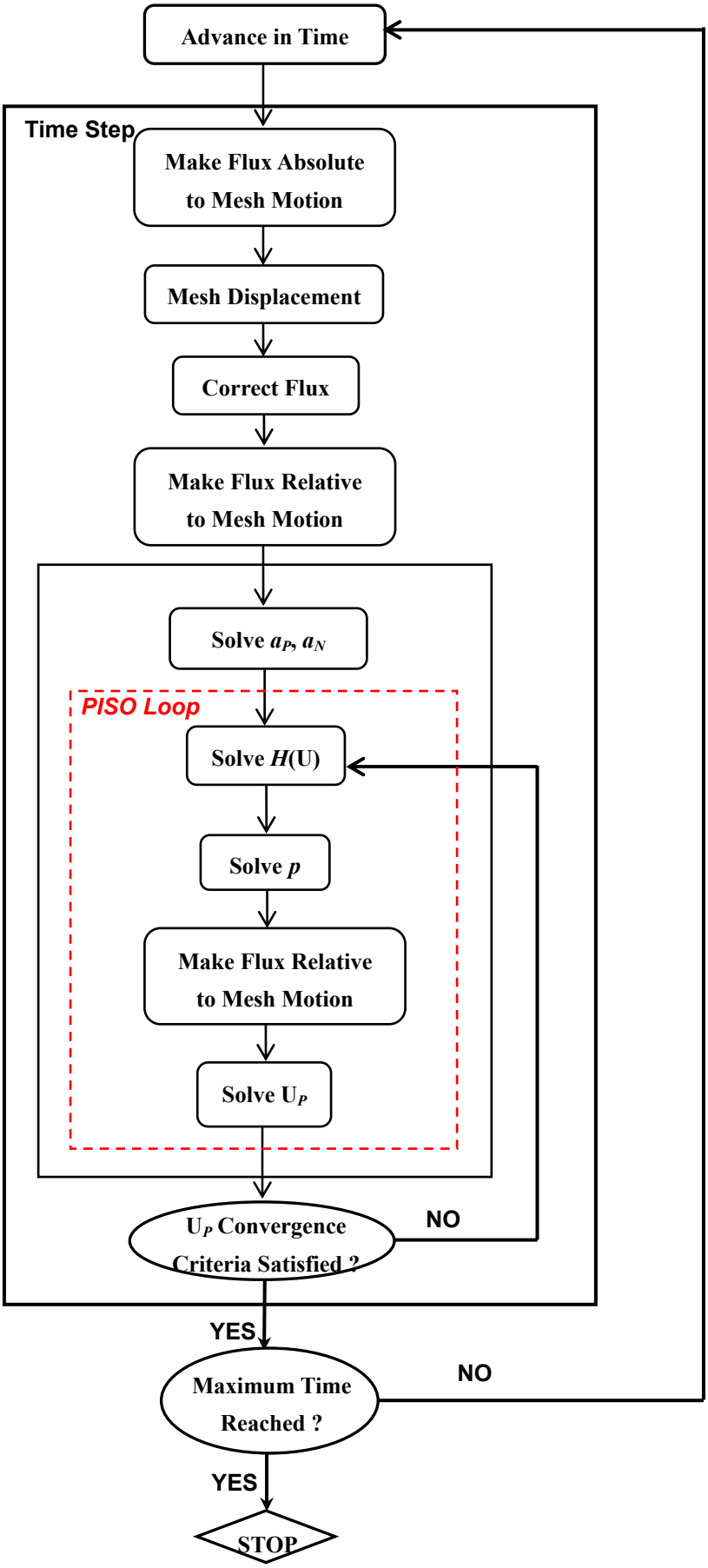


Fig. 4.5 Flow chart of the Sohrab solver for dynamic mesh

5 Applications of the Sohrab Solver

In the chapters before, the discretisation of the modified equation of fluid motion has been discussed in detail, thus establishing the Sohrab solver by employing the open source software OpenFOAM. In this chapter, the numerical simulations of several flow models are presented to explain and evaluate the applications of this novel solver, especially the procedure for deriving the convective velocity at different flow scales. First, an example of laminar boundary layer flow over a flat plate will be computed to examine the accuracy of the Sohrab solver by comparing the obtained numerical results with the analytical solution of the modified equation of fluid motion. Second, the turbulent flow problems will be simulated, including turbulent boundary layer flow over a flat plat and turbulent flow around an airfoil. Third, under different geometric conditions, the computation of turbulent flow in a 3D curved duct will be discussed to confirm the ability of the Sohrab solver to calculate turbulent flow for a 3D case. Fourth, secondary flows in cascades will be simulated to give an initial test to the Sohrab solver in turbomachinery applications. Finally, rotor–stator interaction in centrifugal pumps will be computed to further validate the Sohrab solver. The results are evaluated by comparing them with those obtained by other methods, including the numerical results obtained by the Navier–Stokes solver and measured data reported in literatures. The computational effort and accuracy of the solver are emphasized to highlight the linear advantages of the Sohrab solver as compared with the Navier–Stokes solver.

5.1 Boundary Layer Flow over a Flat Plate

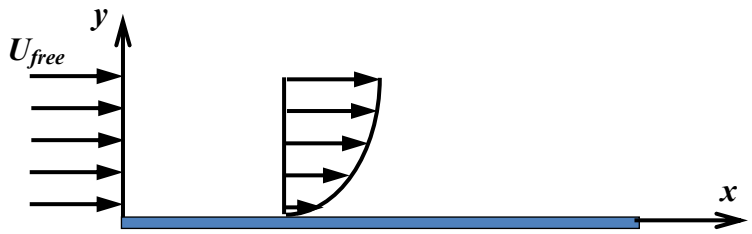


Fig. 5.1 Laminar boundary layer flow over a flat plate

In case of that flow past the steam line body, the influence of viscosity is confined to a very thin layer in the immediate vicinity of the body in which the velocity gradient normal to the wall dU/dy, is very large. As sketched in Fig. 5.1, in this region the component of velocity parallel to the wall increases from zero at the wall to the value U_{free} in the freestream across the boundary layer. The layer under consideration is named as boundary layer, and the concept is due to Prandtl [40]. Inside the boundary layer, the very small viscosity μ of the fluid exerts an essential influence in so far as the shearing stress τ may assume large values. Outside the boundary layer, no large

velocity gradients occur and the influence of viscosity is unimportant. In other words, in this region the flow is frictionless and potential. Studies on boundary layer flows have paved the ways in understanding the wakes or eddies effects on the structure and effective velocity within the zone of boundary layer [6].

As an initial test, the Sohrab solver is applied to simulate this fundamental flow problem to examine the accuracy of the solver and the turbulence assumptions discussed in Chapter 2. First, the Sohrab solver is validated by the analytical solution of the modified equation of fluid motion and the measured data for the laminar boundary layer flow problem. Then, the numerical issues including numerical errors and mesh resolutions are discussed for both Sohrab and Navier–Stokes solvers. Finally, by using the velocity field constructed using the laminar flow as convective velocity, the Sohrab solver is applied for the turbulence boundary conditions.

5.1.1 Laminar Flow

In this section, as the application of the Sohrab solver, an example of the laminar boundary layer flow over a flat plate is computed numerically. Available analytical and experimental results are of considerable importance in verifying the accuracy and validity of numerical results. For this reason, the modified equation of fluid motion is solved for the flow model analytically as well. The experimental results are collected from several literatures, including the classical Blasius [5] and the modern laser measurement [16], to give more reliable comparisons.

5.1.1.1 Analytical Solution

The analytical solution for the laminar boundary layer flow problem has been discussed by Sohrab [4, 9, and 10] and Wan et al. [41]. In this section, this solution is only briefly presented.

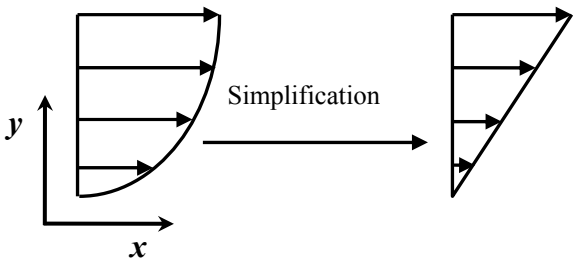


Fig. 5.2 Assumption of the velocity inside the boundary layer

As sketched in Fig. 5.1, the resultant velocity profile inside the boundary layer has a large gradient in y direction. As an assumption, Sohrab [4] simplified this physical velocity profile to a linear one to solve the modified equation of fluid motion analytically for this model. On the basis of the assumption in Fig. 5.2, the average

velocity inside the laminar boundary layer is

$$\bar{U} = \frac{1}{2} U_{free} \tag{5.1}$$

Referring to the convective velocity analysis in Chapter 2, it is found that the convective velocity is equal to the average velocity for this flow model. In other words, outside the boundary layer, one has

$$w_{out} = U_{free} \tag{5.2}$$

while, inside the boundary layer, one has

$$w_{in} = \bar{U} = \frac{1}{2} U_{free} \tag{5.3}$$

It is clear that Eq. (5.3) is required to solve the modified equation of fluid motion for the boundary layer flow problem. Substituting Eq. (5.3) into Eq. (2.30) leads to the solution [4]:

$$U / U_{free} = erf(\xi) \tag{5.4}$$

in terms of the similarity variable

$$\xi = y \frac{\sqrt{U_{free}}}{2\sqrt{2\nu x}} \tag{5.5}$$

Hence the analytical solution of the modified equation of fluid motion for the velocity profile inside the laminar boundary layer can be expressed by an error function. Note that this solution is based on the assumption in Fig. 5.2. This approximated analytical solution will be improved by the numerical solution in the following discussions.

5.1.1.2 Numerical Solution

In order to obtain a smooth entrance flow at the flat plate, an additional section is considered to be added before the flat plate for modeling the laminar boundary layer flow problem. As sketched in Fig. 5.3, a 2D numerical computation model, including the entrance flow section and the flat plate, is built up for the simulation.

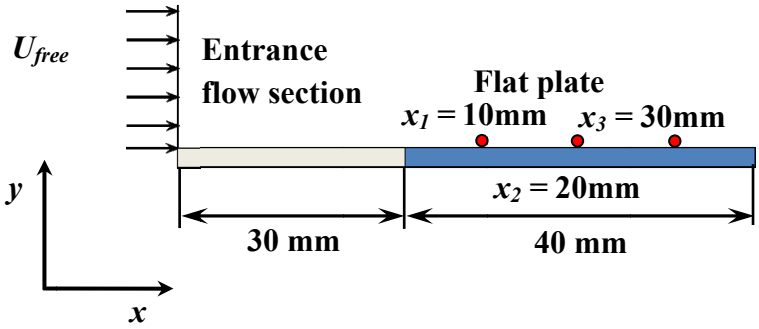


Fig. 5.3 Numerical model of the laminar boundary layer flow over a flat plate

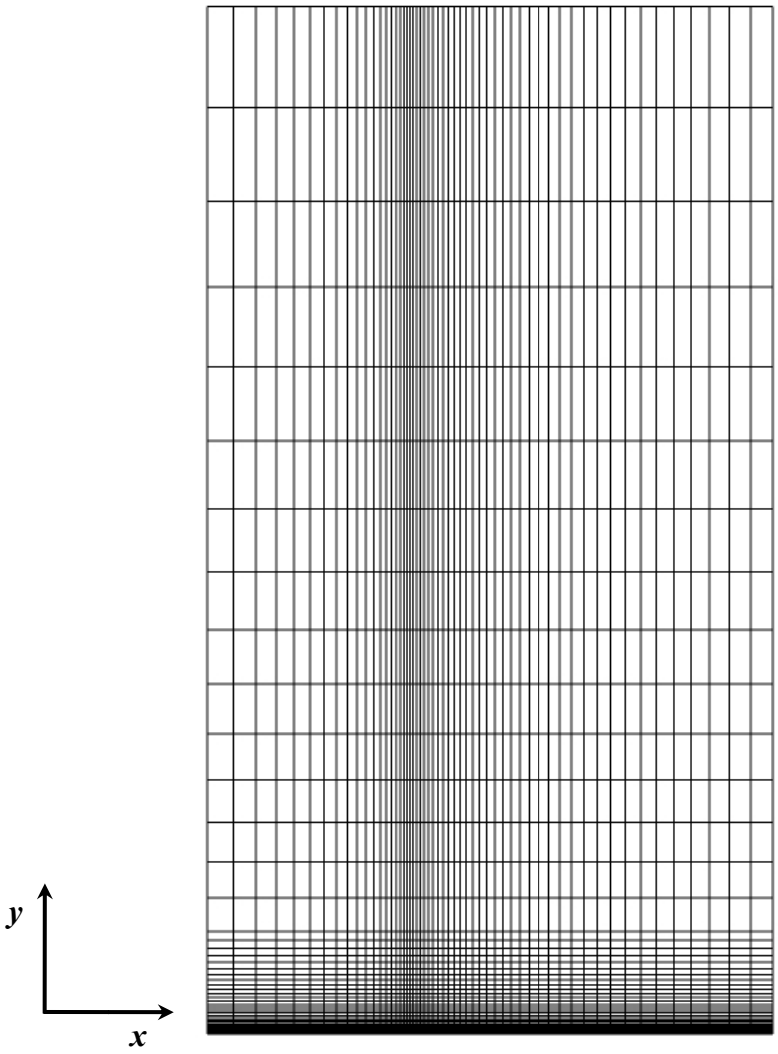


Fig. 5.4 Grid for the laminar flow over a flat plate

Corresponding to the numerical model, the 2D structured grids for the computational domain contains both entrance flow section and the flat plate, as shown in Fig. 5.4. A grid–independent study is conducted for the model, and the grids number for the whole computational domain is fixed to 5,000. In order to produce a smooth velocity profile

inside the laminar boundary layer, a finely graded mesh is plotted next to the flat plate, followed by a more coarsely graded mesh above. Note that the top of the mesh is far from the flat plate, for reason of a satisfactory Neumann boundary condition. For boundary conditions, the Reynolds number is limited to obtain a laminar flow in the whole computational domain. At the inlet the freestream velocity and the convective velocity (Eq. (5.3)) are defined, while at the outlet, the average static pressure is given. At the entrance flow section, Neumann boundary condition is used to provide a smooth inlet flow for the followed flat plate. At the flat plate, no slip boundary condition is used. Details about the boundary conditions are summarized in Tab. 5.1.

Inlet	U_{free} = 0.2 m/s, w = 0.1 m/s
Outlet	Average static pressure p = 0 Pa
Entrance flow section	Neumann B. C.
Flat plate	No–slip B. C.
Working fluid	$v = 1.0\times10^{-6}$ m^2/s
Reynolds number	Re = 1,100, laminar flow

Tab. 5.1 Boundary conditions setup for the laminar boundary layer flow

The boundary layer flow over a flat plate can be considered as a steady flow process. Therefore the time–dependent component in the modified equation of fluid motion (Eq. (2.30)) can be neglected and the derivation of the equation becomes

$$\mathbf{w}_\beta \cdot \nabla \mathbf{U}_\beta - \nu_\beta \nabla^2 \mathbf{U}_\beta = -\frac{\nabla p_\beta}{\rho_\beta} \tag{5.6}$$

In the numerical issues setup, the Gaussian quadrature is applied for the finite volume discretisation of the whole equation and different schemes are assigned to the detailed components. As shown in Tab. 5.2, the convection component is solved by a first order accurate differencing scheme UD, while the others are solved by the second order accurate schemes.

Component:	**Scheme:**	**Numerical behaviour:**
Convection	UD	First order, bounded
Diffusion	CD	Second order, unbounded
Pressure gradient	CD	Second order, Gaussian integration

Tab. 5.2 Numerical schemes setup for the laminar boundary layer flow

The resultant velocity profiles at three x–axis positions as shown in Fig. 5.3 are plotted in y direction to show the boundary layer development. As presented in Fig. 5.5, all

three represent the characteristics of the boundary layer flow in physics, increasing from zero at the wall to the value U_{free} in the freestream across thin layers. In addition, the thickness of the boundary layer increases in the flow direction. To validate the results obtained by the Sohrab solver, more comparisons are required.

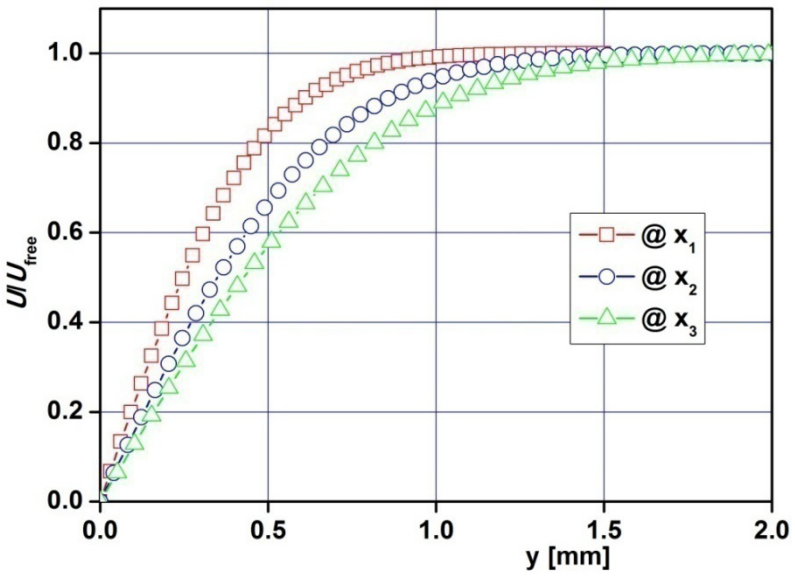


Fig. 5.5 Calculated velocity profiles inside the laminar boundary layer

5.1.1.3 Results & Comparisons

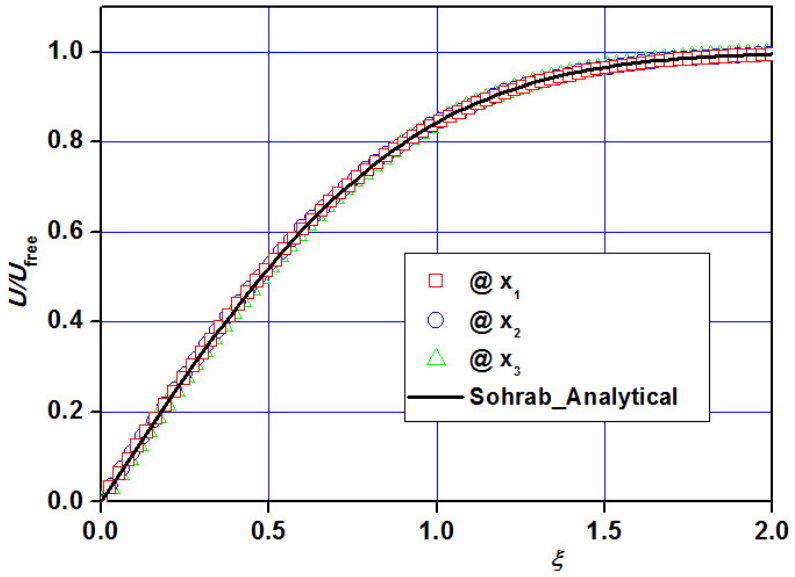


Fig. 5.6 Comparison of the velocity profile inside the laminar boundary layer between the numerical and analytical solutions of the modified equation of fluid motion

The modified equation of fluid motion has been solved for the laminar boundary layer problem by both analytical and numerical methods. In this section, the obtained

results are compared with the classical Navier–Stokes solutions and the measured data to give a validation to the modified equation of fluid motion.

Using the similarity variable in Eq. (5.5), the three velocity profiles plotted in Fig. 5.5 are compared with the analytical solution in Eq. (5.4). As shown in Fig. 5.6, all the three numerical resultant profiles follow the analytical solution exactly. This result demonstrates that the Sohrab solver developed in the present work can precisely reproduce the analytical solution by the numerical method for the modified equation of fluid motion.

Noting the simplification in Fig. 5.2, it is found that the solution for the convective velocity (Eq. (5.3)) may be slightly deflective because of the linear assumption. As discussed by Sohrab [10], the linear assumption could be improved by averaging the resultant velocity profile in Fig. 5.6, making a new average convective velocity and solving the modified equation again for the velocity. This process can be iterated by using the developed numerical code. When the solution converges, the new result of the convective velocity is

$$w'_{in} = 0.606U_{free} \tag{5.7}$$

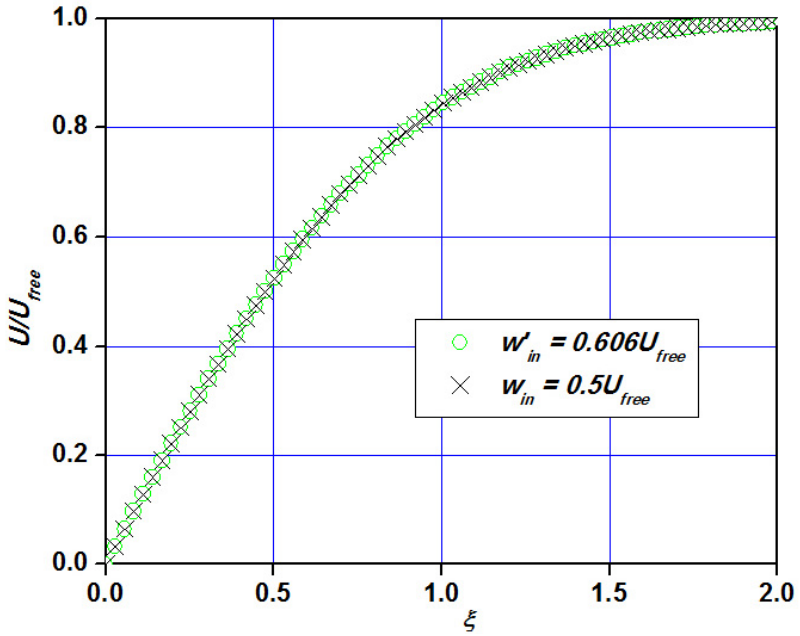


Fig. 5.7 Comparison of the velocity profile inside the laminar boundary layer between the numerical solutions derived from different convective velocities

Then the updated new velocity profile is compared with the old result which is under the assumption of Eq. (5.3). As presented in Fig. 5.7, after the convective velocity iteration is carried out, the updated velocity profile is almost identical to the old one.

This comparison demonstrates that the convective velocity deviation influence caused by the assumption in Fig. 5.2 is negligible for the modified equation of fluid motion and the convective velocity defined in Eq. (5.3) satisfies both analytical and numerical solutions.

Using the grids in Fig. 5.4, the boundary conditions in Tab. 5.1 and the numerical schemes in Tab. 5.2, the Navier–Stokes solver is also applied to simulate this laminar flow model.

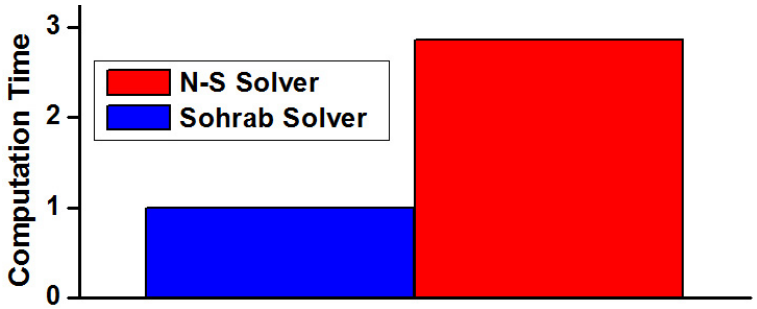


Fig. 5.8 Comparison of the computation time taken by both solvers

The computation time of the two solvers is compared in Fig. 5.8. Obviously, the Sohrab solver consumes much less time than the classical Navier–Stokes solver. This result offers the evidence in support of the analysis in Chapter 3 and Chapter 4. The developed iteration technique in the Sohrab solver exerts the linear advantage of the modified equation of fluid motion to accelerate the numerical solution.

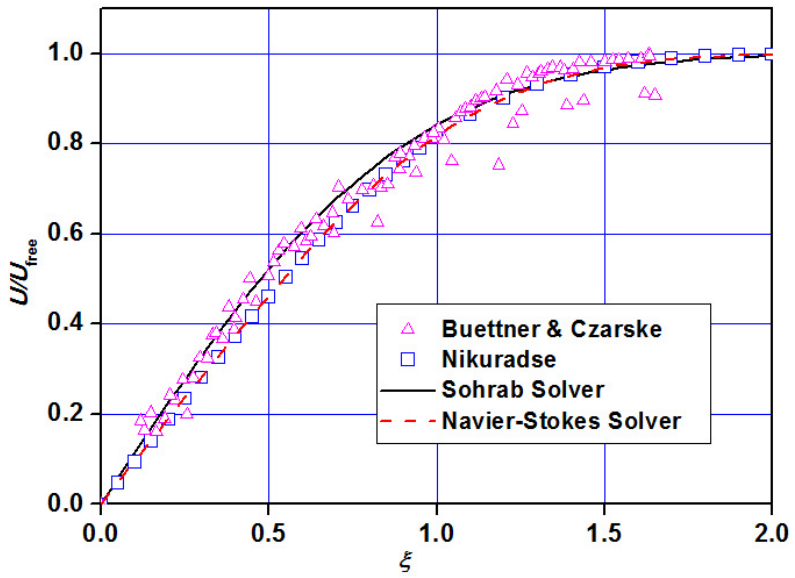


Fig. 5.9 Comparison of the velocity profile inside the laminar boundary layer between the experiment and numerical solutions

The resultant velocity profiles obtained by both numerical solvers are compared with the experimental data including the well known results of Nikuradse [17] and the recent laser Doppler interferometry measurement of Büttner and Czarske [16]. As shown in Fig. 5.9, the prediction of the Sohrab solver has a close agreement with the other three velocity profiles. In detail, the Sohrab solver result conforms more closely to the measured data of Büttner and Czarske [16], while the Navier–Stokes solver result exactly follows that of Nikuradse [17]. In the laser experimental results reported by Büttner and Czarske [16], deviation of the few velocity points from the other results can be found. As discussed by Büttner and Czarske [16], this deviation is caused by an obstacle (a wire with 1 mm diameter) which was placed upstream on the glass plate and induced a turbulence.

Consequently, it is found that the Sohrab solver indeed provides a numerical solution with a satisfactory accuracy for this laminar boundary layer flow problem. In addition, for this flow model, more interesting comparisons between the modified equation results and the diverse measured data can be found in Sohrab [10].

5.1.1.4 Numerical Error Comparisons

The Sohrab solver, as compared to the Navier–Stokes solver, takes much less computation time and provides satisfactory results for the present simulation. In order to explain this result, the numerical errors induced by the numerical schemes and mesh resolutions in the Sohrab solver results are investigated by comparing them with those obtained from the Navier–Stokes solver.

As discussed in Chapter 3, the difference of the numerical error between the two solvers arises from the numerical diffusion. Taking the method UD ($\gamma = 1$) on a uniform mesh ($f_x = 1/2$), the results for Eq. (3.38) and Eq. (3.39) can be obtained as follows:

$$\Gamma_{Num\_N\text{-}S} = \frac{1}{2} \cdot \mathbf{U}_f \cdot PN \tag{5.8}$$

$$\Gamma_{Num\_M} = \frac{1}{2} \cdot \mathbf{w}_f \cdot PN \tag{5.9}$$

Then the simulation parameters are varied to evaluate the influence caused by these numerical errors on the results obtained from the two solvers. In terms of the parameter variation, the numerical tests for six associated cases, as listed in Tab. 5.3, are performed. For simplicity, the Reynolds number is preserved as a constant in the laminar flow range, by varying the kinematic viscosity. Since the convective velocity is calculated by the inlet velocity (Eq. (5.3)), the numerical error can be enlarged by increasing the inlet velocity for both solvers, comparing cases 1 and 2 with cases 3 and 4. On the other hand, the error also linearly depends on the grid spacing *PN*, then,

Case 1:

$U_{free} = 0.2$ [m/s]	$\nu = 1.0\times10^{-6}$ [m^2/s]	$PN = 1.0\times10^{-4}$ [m]	Scheme: UD
$\Gamma_{Num\_N\text{-}S(max)} = 1.0\times10^{-5}$ [m^2/s]		$\Gamma_{Num\_M} = 0.5\times10^{-5}$ [m^2/s]	

Case 2:

$U_{free} = 0.2$ [m/s]	$\nu = 1.0\times10^{-6}$ [m^2/s]	$PN = 1.0\times10^{-4}$ [m]	Scheme: CD
$\Gamma_{Num\_N\text{-}S(max)} = 0$ [m^2/s]		$\Gamma_{Num\_M} = 0$ [m^2/s]	

Case 3:

$U_{free} = 2$ [m/s]	$\nu = 1.0\times10^{-5}$ [m^2/s]	$PN = 1.0\times10^{-4}$ [m]	Scheme: UD
$\Gamma_{Num\_N\text{-}S(max)} = 1.0\times10^{-4}$ [m^2/s]		$\Gamma_{Num\_M} = 0.5\times10^{-4}$ [m^2/s]	

Case 4:

$U_{free} = 2$ [*m/s*]	$\nu = 1.0\times10^{-5}$ [m^2/s]	$PN = 1.0\times10^{-4}$ [m]	Scheme: CD
$\Gamma_{Num\_N\text{-}S(max)} = 0$ [m^2/s]		$\Gamma_{Num\_M} = 0$ [m^2/s]	

Case 5:

$U_{free} = 2$ [m/s]	$\nu = 1.0\times10^{-5}$ [m^2/s]	$PN = 1.0\times10^{-5}$ [m]	Scheme: UD
$\Gamma_{Num\_N\text{-}S(max)} = 1.0\times10^{-5}$ [m^2/s]		$\Gamma_{Num\_M} = 0.5\times10^{-5}$ [m^2/s]	

Case 6:

$U_{free} = 2$ [m/s]	$\nu = 1.0\times10^{-5}$ [m^2/s]	$PN = 1.0\times10^{-5}$ [m]	Scheme: CD
$\Gamma_{Num\_N\text{-}S(max)} = 0$ [m^2/s]		$\Gamma_{Num\_M} = 0$ [m^2/s]	

Tab. 5.3 Boundary conditions of the test cases

the numerical error is rolled back by a much finer mesh for cases 5 and 6. It is noticeable from the discussion in sections 3.3 and 4.1.1, that the increasing of the inlet velocity always enlarges the boundedness problem in the space discretisation. Considering this, both discretisation schemes UD and CD are applied to investigate the boundedness limit in the numerical solutions of the two solvers.

Comparing Γ_{Num} between the Sohrab (indexed by *M*) and Navier-Stokes solvers, it is found that $\Gamma_{Num\_N\text{-}S}$ in the Navier–Stokes solver solution experiences variations induced by the velocity profile inside the boundary layer, thus resulting in the maximum $\Gamma_{Num\_N\text{-}S}$ at the top of the boundary layer, while Γ_{Num_M} in the Sohrab solver solution is preserved as a constant for the whole boundary layer because of the fixed convective velocity. As shown in Fig. 5.10, Γ_{Num_M} in the Sohrab solver solution is even smaller than the averaged $\Gamma_{Num\_N\text{-}S}$ in the Navier–Stokes solution. This result makes strong influence to the accuracy of the solvers.

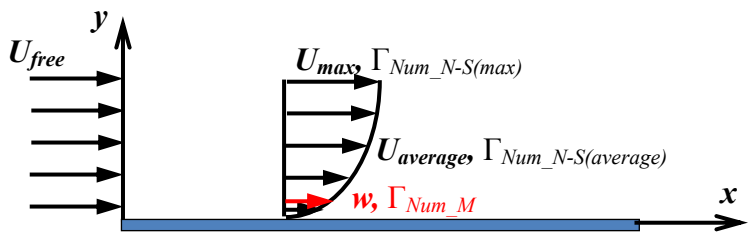


Fig. 5.10 Comparison of the magnitude of the convective velocity and boundary layer velocities

First, the influence caused by the discretisation schemes is under discussion. The results show that for the Navier–Stokes solver the discretisation scheme CD can only be applied to the cases 2 and 6, since it terminates the case 4 during the solver running. The reason arises from the boundedness analysis in sections 3.3 and 4.1.1. However, for the Sohrab solver all three cases give the same results. Then, the results of the cases 1, 2, 3, 5 and 6 obtained by both Sohrab and Navier–Stokes solvers are compared. As shown in Fig. 5.11, when the numerical error is small enough (cases 1 and 2), both solvers can provide satisfactory results. However when the inlet velocity increases, the numerical diffusion severely distorts the Navier–Stokes solution and this problem can only be balanced by an excessively fine mesh as case 5 or 6. In comparison, the discretisation in the Sohrab solver results a relative small and stable numerical error for the whole computational domain, thus even accepting a lower quality mesh to complete the numerical simulation and produce the results with satisfactory accuracy. The stable numerical error in the Sohrab solver explains the computation time difference between the two solvers shown in Fig. 5.8. In addition, the convergence speeds of the two solvers can also be imagined in a large difference. This issue will be discussed in the following sections with other flow models.

In the present numerical error test, the convective velocity expressed by Eq. (5.3) makes the numerical diffusion is easier to control for the Sohrab solver as illustrated in Fig. 5.10. However this special case may not stand for the complicated flow boundary conditions. Consequently, another numerical error test is presented for the turbulent secondary flow in a 3D curved duct, for which a more complex convective velocity is constructed from the laminar secondary flow field.

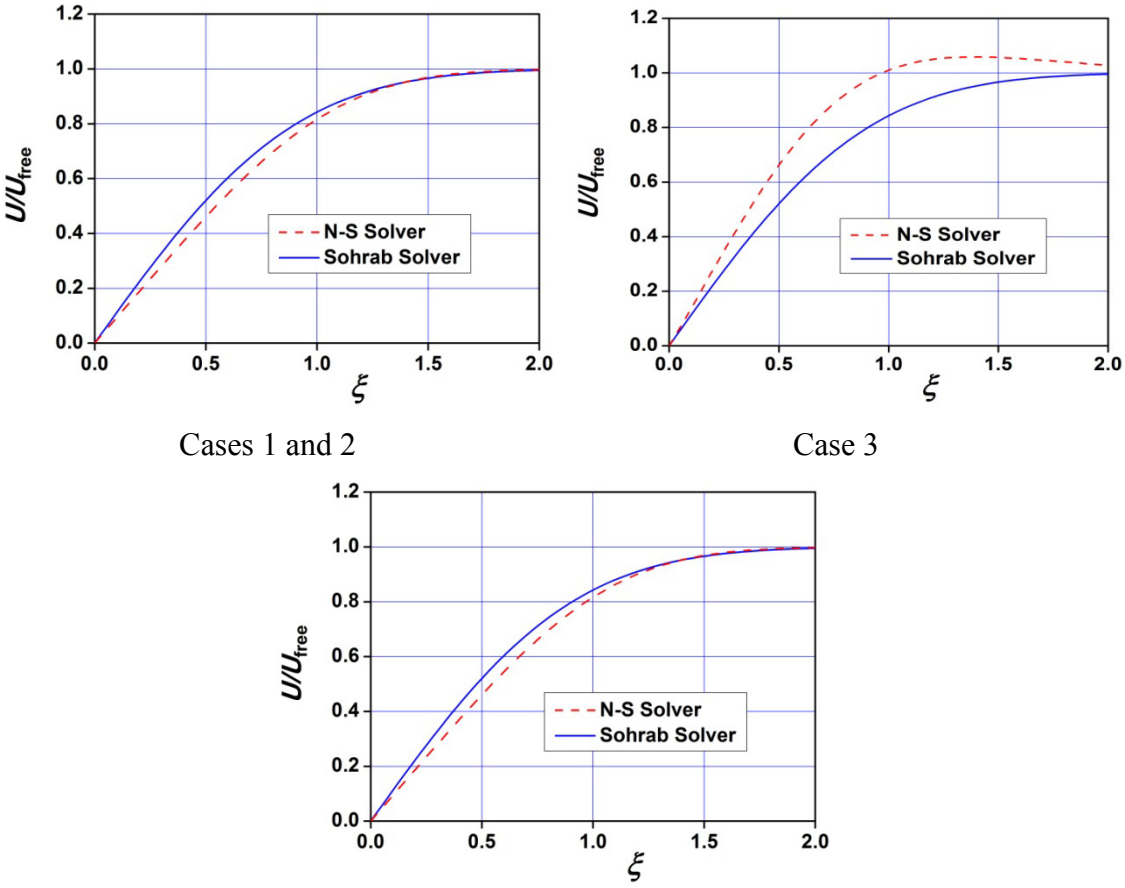


Fig. 5.11 Numerical error effect on velocity profiles

5.1.2 Turbulent Flow

This section discusses the application of the Sohrab solver for the turbulent flow over a flat plate. As compared to the laminar flow application, the Reynolds averaged modified equation of fluid motion (Eq. (2.50)) is solved for the turbulent boundary layer flow by using a turbulence model in the Sohrab solver. It is emphasized that the convective velocity for the turbulence scale is constructed from the laminar flow simulation. The obtained results are validated by those obtained by the Navier–Stokes solver and measured data in literatures.

At a high Reynolds number $Re = 2.03\times10^6$, the turbulent boundary layer flow over a flat plate is simulated by the Sohrab solver. As compared to the model in Fig. 5.3, a much longer plate, as sketched in Fig. 5.12, is considered to achieve the high Reynolds number in the numerical simulation.

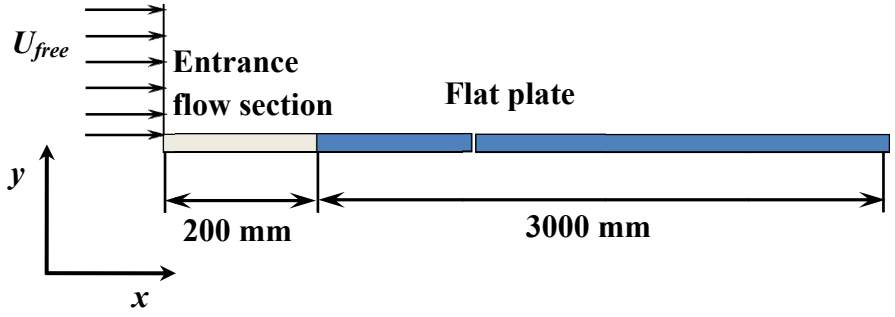


Fig. 5.12 Numerical model of the turbulent boundary layer flow over a flat plate

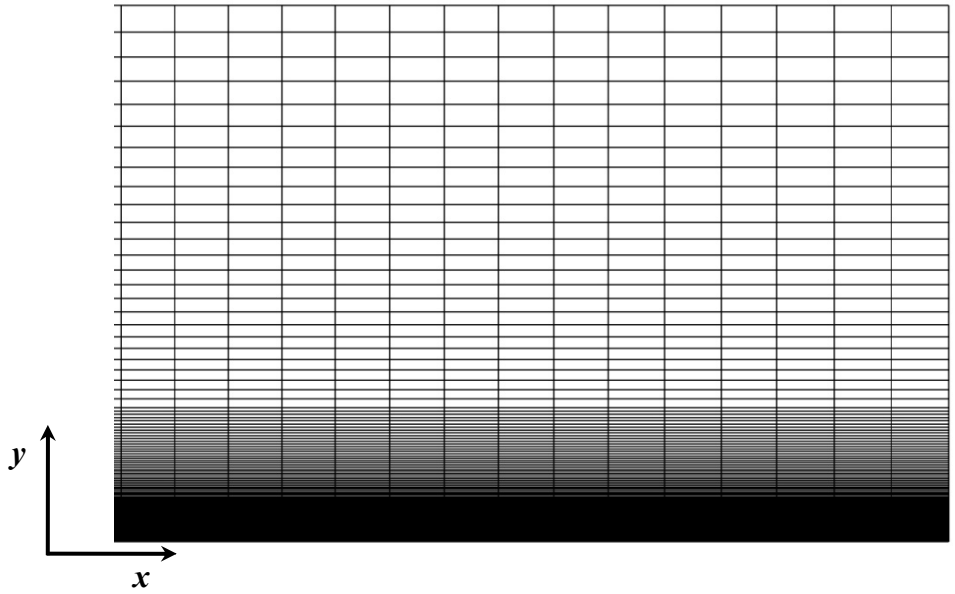


Fig. 5.13 Grid for the turbulent flow over a flat plate

In the Sohrab solver, the turbulence transient feature is solved by the turbulence model. In order to reduce the numerical error caused by the mesh resolution, the y^+ value calculated from the turbulence model should be below 5. The parameter y^+ is considered to give a valuable evaluation to the mesh and wall function in the turbulent flow. It is the dimensionless distance between the first node of the mesh and wall:

$$y^+ = \frac{\Delta y\sqrt{\tau_w \rho}}{\mu} \tag{5.10}$$

where Δy is the normal distance between the first node and τ_w is the wall shear stress.

Consequently, a higher quality mesh is required for the present simulation. Part of the mesh is shown in Fig. 5.13. The first grid node next to the plate is well controlled in the mesh to make the maximum y^+ below 4.

By using the velocity profile inside the laminar boundary layer as the convective velocity (plotted in Fig. 5.6), the Sohrab solver is applied to the simulation. The detailed boundary conditions are listed in Tab. 5.4 and the numerical schemes are the same as those in Tab. 5.2.

Inlet	U_{free} = 19.4 m/s, $w = U_{laminar}$, 5% turbulence intensity
Outlet	Average static pressure p = 0 Pa
Entrance flow section	Neumann B. C.
Flat plate	No–slip B. C.
Working fluid	$\nu = 1.43\times10^{-5}$ m^2/s
Reynolds number	$Re = 2.03\times10^6$, turbulent flow
Turbulence model	k–ε

Tab. 5.4 Boundary conditions setup for turbulent boundary layer flow

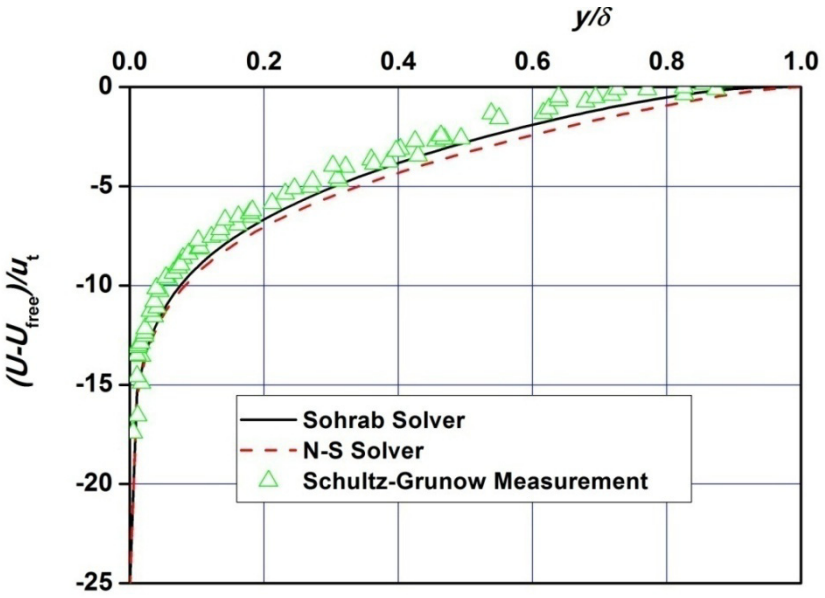


Fig. 5.14 Comparison of the velocity profile inside the turbulent boundary layer

Resultant velocity profiles inside the turbulent boundary layer are compared with the Navier–Stokes results and the experimental data reported in [42]. As shown in Fig. 5.14, the complete velocity profile shape constructed using the Sohrab solver is slightly higher than that constructed using the Navier–Stokes solver, and the entire shape of the former exactly coincides with the latter. In terms of the measured data, the numerical results obtained from the both solvers have satisfactory agreement with the measured

data points. Thus, this result confirms the accuracy of the Sohrab solver in calculating turbulent flow. Furthermore, it evidences the discussions of the Reynolds averaged modified equation of fluid motion and the derivation procedure of the convection velocity for turbulence scale which is presented in section 2.4.

5.2 Flow around an Airfoil

On the basis of the success for simulating the fundamental boundary layer flow problems, the Sohrab solve is applied to carry out numerical simulations for the flow around an airfoil under both laminar and turbulent flow environments. According to Fig. 2.4, a numerical method to construct the convective velocity for both laminar and turbulent flows is presented. The obtained results are evaluated by comparing them with those obtained by the classical Navier–Stokes solver and valid measured data.

5.2.1 Model

The airfoil model used for the simulation was obtained from the experiments of Thom and Swart [43]. The geometric profile of the model as shown in Fig. 5.15 is a modified version of the classical 2D Royal Air Force (RAF) 6, which has square ends on both sides because of poor manufacturing techniques practiced in 1940, when this airfoil was first constructed.

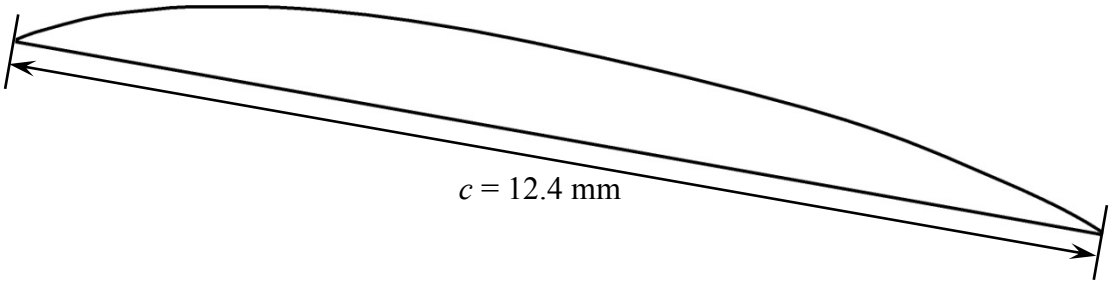


Fig. 5.15 Geometry profile of the airfoil RAF 6

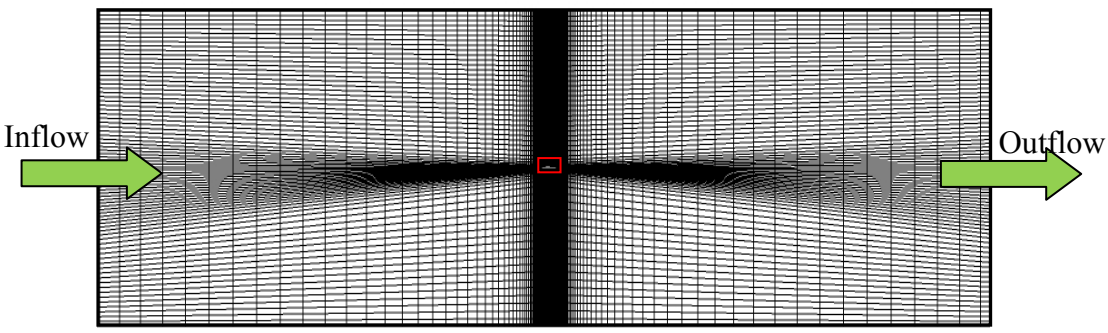


Fig. 5.16 Complete grid region for the airfoil at $\alpha = 0°$

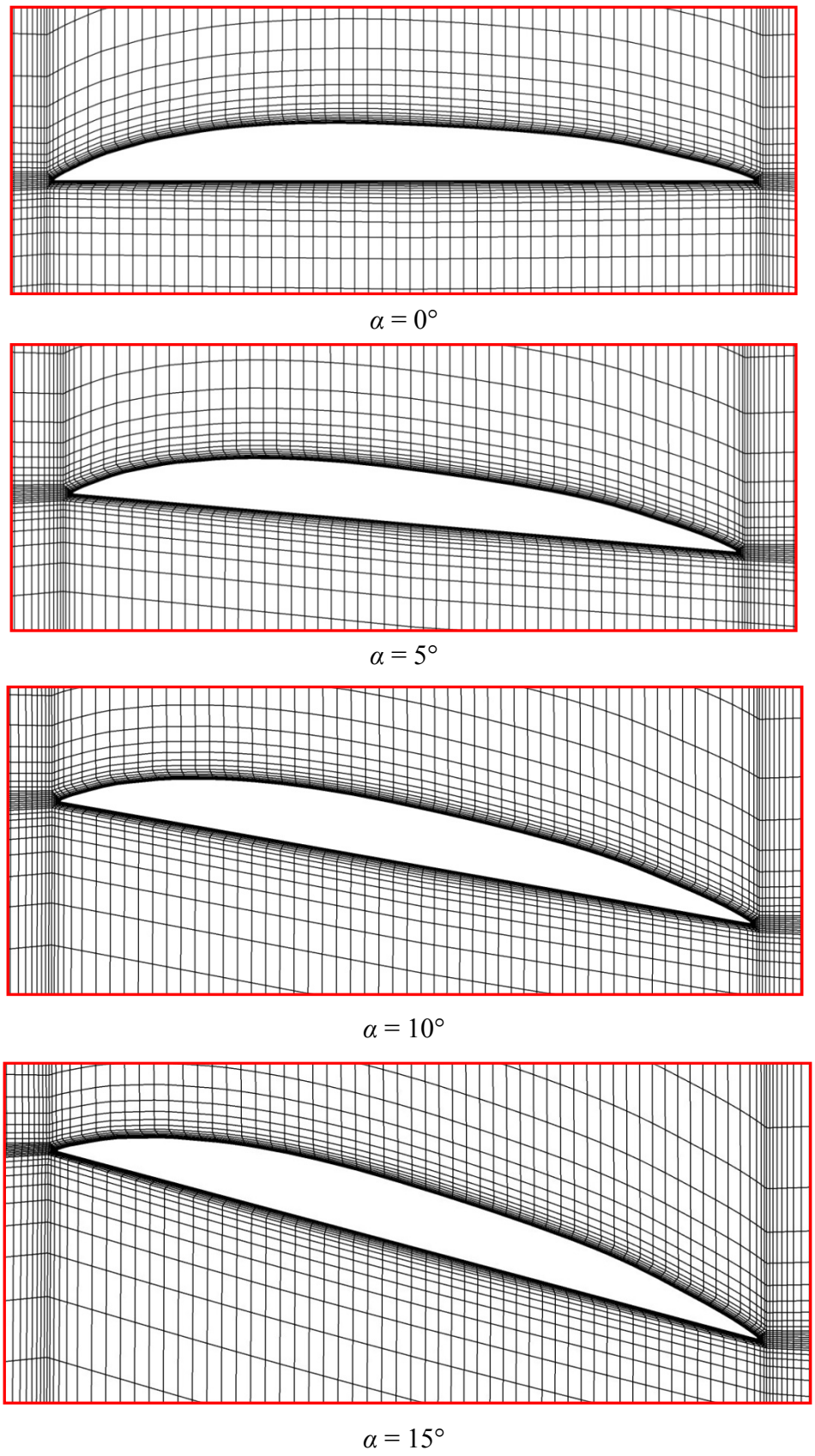


Fig. 5.17 Detailed grid for the airfoil at four angles of attack

A 2D mesh is built up for the numerical simulation. The entire calculation region with a total grid number of about 58,000 is shown in Fig. 5.16. Corresponding to different angles of attack, four mesh models whose detailed structures around the airfoil are shown in Fig. 5.17, are developed for the numerical simulations discussed further.

5.2.2 Laminar Flow

First, the laminar flow around the airfoil is under consideration. Both Sohrab and Navier–Stokes solvers are applied to simulate the flow with the same mesh, boundary conditions and numerical schemes setup, thus resulting in an equivalent access for the results comparisons. In addition, the measured data from [43] are presented in the results comparisons for validations.

5.2.2.1 Numerical Simulation Setup

As explained in Fig. 2.4, the velocity in the potential flow field is taken as convective velocity in the laminar flow field. At potential flow scale, Eq. (2.33) indicates the relation between the modified equation of fluid motion and the scale–invariant form of the Euler equation. Using this equation, $\mathbf{w}_{laminar}$ is directly solved by the Euler equation. Consequently, for the present airfoil model, the convective velocity field can be constructed by executing potentialFOAM, which is a special solver available with the OpenFOAM application for solving only Euler equations.

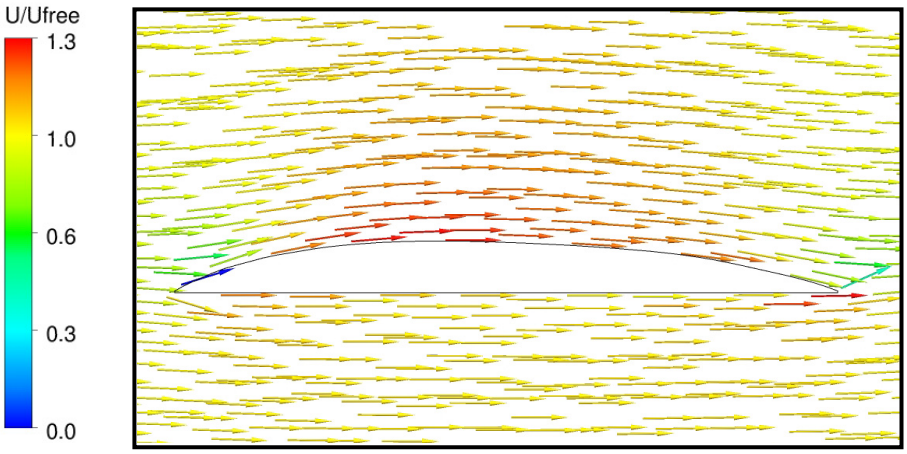


Fig. 5.18 Potential flow around the airfoil

As shown in Fig. 5.18, the calculated convective velocity field simply follows the airfoil profile, and any viscous effect is ignored. The variation in the velocity magnitude is caused by the effect of the continuity equation.

For other boundary conditions, the Reynolds number is limited in an ultra low range to

obtain laminar flow in the whole computational domain. At the inlet an inlet velocity and the aforementioned convective velocity are defined, while at the outlet, the average static pressure is given. At the airfoil, no slip boundary condition is used. Due to the 2D properties, all symmetric walls are defined as symmetric boundary conditions. Details about the boundary conditions are summarized in Tab. 5.5.

Inlet	$\mathbf{U}_{in}$, $\mathbf{w} = \mathbf{U}_{potential}$
Outlet	Average static pressure $p = 0$ Pa
Airfoil	No–slip B. C.
Symmetric walls	Symmetric B. C.
Working fluid	$\nu = 1.1\times10^{-6}$ m^2/s
Reynolds number	$2 \le Re \le 15$, laminar flow

Tab. 5.5 Boundary conditions setup for the laminar flow around an airfoil

The steady state equation (Eq. (5.6)) is solved for the steady laminar flow around the airfoil by the Sohrab solver. Because the Reynolds number is ultra low, the second order accurate scheme, linear scheme is used for the discretisation of the convection component. As discussed in Chapter 3, this scheme preserves the discretisation in a high accuracy by avoiding the numerical diffusion error. The details about numerical schemes are listed in Tab. 5.6.

Component:	**Scheme:**	**Numerical behaviour:**
Convection	CD	Second order, unbounded
Diffusion	CD	Second order, unbounded
Pressure gradient	CD	Second order, Gaussian integration

Tab. 5.6 Numerical schemes setup for the laminar flow around an airfoil

5.2.2.2 Results & Comparisons

In this section, the results comparisons are presented by using the drag coefficient. The drag coefficient can quantify the drag of the airfoil in a fluid environment, thus indicating the aerodynamic characteristics of an airfoil. This dimensionless coefficient is defined as:

$$C_d = \frac{2\mathbf{F}_d}{\rho c l \mathbf{U}^2} \tag{5.11}$$

where $\mathbf{F}_d$ is the drag force which is defined by the force component in the direction of the flow, c is the chord of the airfoil and l is the span.

Using this definition, the calculated drag coefficients versus different Reynolds numbers at an angle of attack $\alpha = 0°$ obtained using both Sohrab and Navier–Stokes solvers are compared with the experimental data, as shown in the following figure:

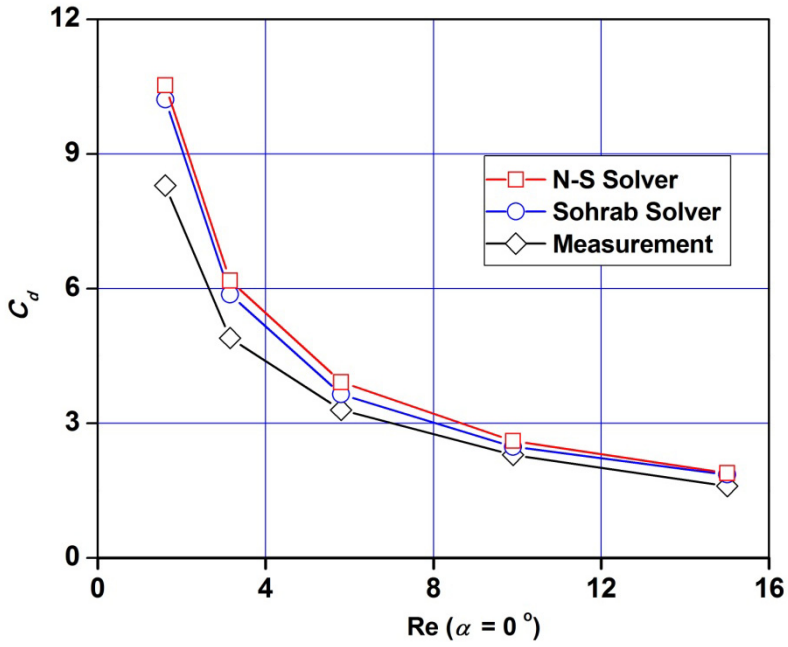


Fig. 5.19 Comparison of the drag coefficient for diverse Reynolds numbers

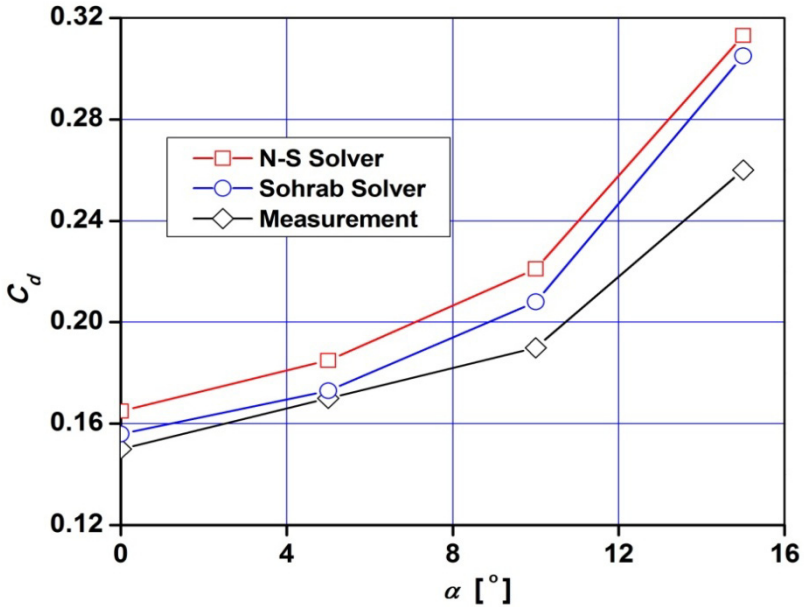


Fig. 5.20 Comparison of the drag coefficient for different angles of attack

As shown in Fig. 5.19, the numerical results of the drag coefficients obtained by the Sohrab solver simply coincide with that obtained by the classical Navier–Stokes solver and exhibit the same trend as that exhibited by the results obtained by the

Navier–Stokes solver: the drag coefficient reduces with an increasing in the Reynolds number. A difference is found between the experimental data and the estimated numerical results, and this difference decreases as the Reynolds number increases. The phenomenon observed here has remained unsolved in the case of numerous numerical CFD codes. For a low Reynolds number, the numerical results of the drag coefficient are always higher than the experimental results.

Similarly, Fig. 5.20 illustrates the variations in the drag coefficient versus different angles of attack. The simulation results shown in this figure are obtained for several Reynolds numbers because each angle in the experiment coincided with a particular Reynolds number. Fig. 5.19 and Fig. 5.20 clearly show that the results calculated by the Sohrab solver are in good agreement with those calculated by the classical Navier–Stokes solver and the experimental results. However, the experimental data greatly differ from the numerical results at very low Reynolds numbers and large angles of attack. This difference can be attributed to the aforementioned numerical reasons and the measurement error introduced into the experiment owing to poor manufacturing techniques practiced in 1940.

5.2.3 Turbulent Flow

By using the velocity field constructed using the aforementioned laminar flow as convective velocity, the Sohrab solver is applied for the airfoil model under turbulence boundary conditions.

k–*ε* turbulence model is applied to simulate the turbulence. Calculating the y^+ of the mesh used in the laminar flow simulation, a very small value is obtained near the airfoil, $y^+ < 8$. This result proves that the aforementioned mesh owns high quality for turbulent flow simulation. Hence, the meshes shown in Fig. 5.16 and Fig. 5.17 are applied. For boundary conditions, the Reynolds number is increased to 100,000 and water at 25 °C is used as the media. The details are listed in Tab. 5.7.

Inlet	$\mathbf{U}_{in}$ = 10.81 m/s, $\mathbf{w} = \mathbf{U}_{laminar}$, 5% turbulence intensity
Outlet	Average static pressure p = 0 Pa
Airfoil	No–slip B. C.
Symmetric walls	Symmetric B. C.
Working fluid	$\nu = 1.34\times10^{-6}$ m^2/s
Reynolds number	$Re = 1.0\times10^5$, turbulent flow
Turbulence model	*k*–*ε*

Tab. 5.7 Boundary conditions setup for the turbulent flow around an airfoil

As discussed in sections 3.3 and 5.1.1.4, the flow model owning a high inlet velocity requires a bounded numerical scheme to reduce the numerical oscillation. Therefore, for the present turbulent flow case, the unbounded scheme CD is not suggested for the Navier–Stokes solver. However, the numerical oscillation can be damped during the Sohrab solver iteration because a fixed convective velocity is presented in the simulation. Consequently, the Sohrab solver is free in the numerical scheme selection, while the Navier–Stokes solver has to choose a bounded scheme which increases the numerical diffusion, when simulating the flow models with high inlet velocity. As a result, instead of CD, the scheme TVD is selected for the Navier–Stokes solver for the present simulation and the scheme CD is applied for the Sohrab solver. The details of the numerical schemes for the two solvers are listed in Tab. 5.8.

Component:	**Scheme:**	**Numerical behaviour:**
Convection	CD/TVD	Second order, unbounded/bounded
Diffusion	CD	Second order, unbounded
Pressure gradient	CD	Second order, Gaussian integration

Tab. 5.8 Numerical schemes setup for the turbulent flow around an airfoil

Because no experimental data is available, only the numerical results obtained by the Sohrab and Navier–Stokes solvers are compared.

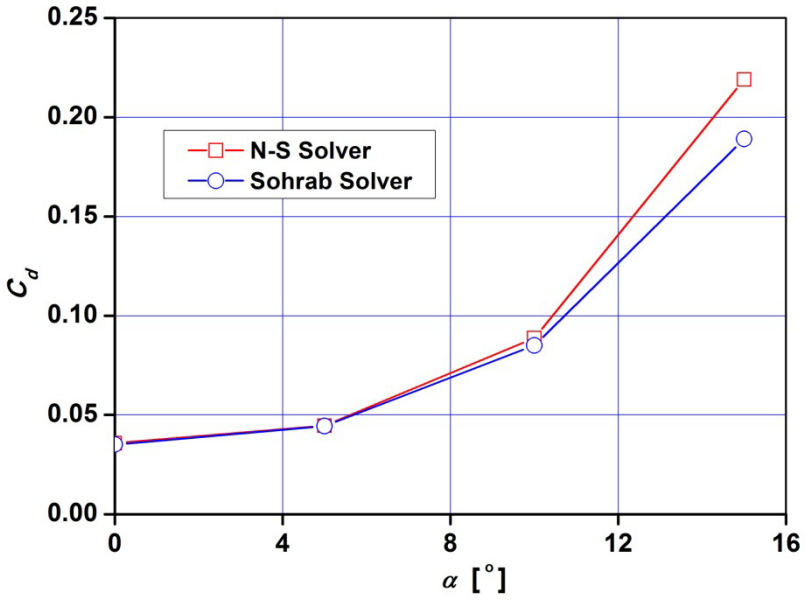


Fig. 5.21 Comparison of the drag coefficient

The Reynolds number is set to $Re = 100{,}000$, and water at 25 °C is used as the media for carrying out the simulation. Various angles of attack are calculated to compare different results obtained using the two numerical solvers.

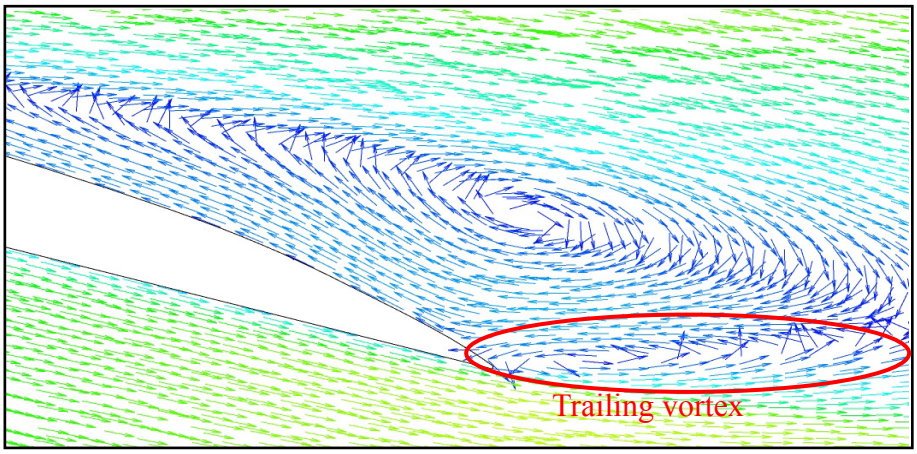


Fig. 5.22 Flow field near the trailing edge calculated by the Navier–Stokes solver

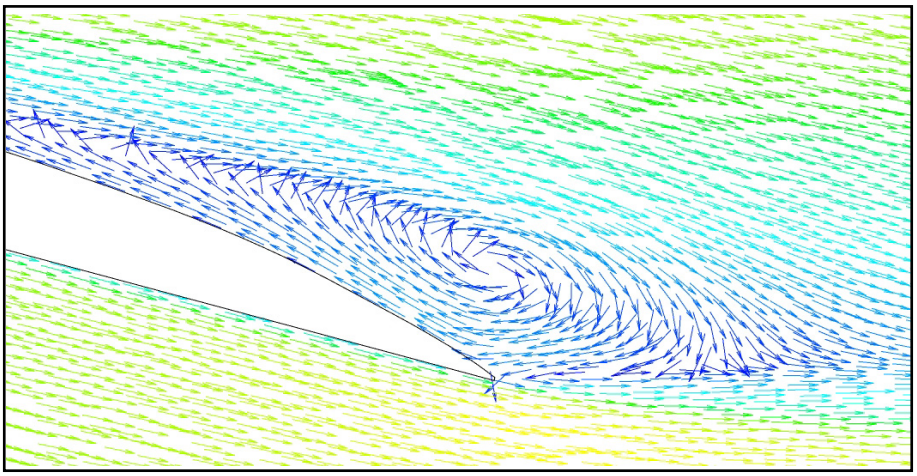


Fig. 5.23 Flow field near the trailing edge calculated by the Sohrab solver

The drag coefficients calculated for different angles of attack by the two solvers are compared. As shown in Fig. 5.21, below $\alpha = 10°$, the curves of the results obtained by the two solvers correspond well with each other; however, above $\alpha = 10°$, the difference increases with increasing angle, and the difference reaches a maximum of 8% at $\alpha = 15°$. The potential reason for this difference can be the flow separation on the airfoil surface and vortex downstream. As shown in Fig. 5.22 and Fig. 5.23, for the solutions obtained using both solvers at $\alpha = 15°$, a vortex can be observed in the flow field. However, the solution obtained by the Navier–Stokes solver shows an additional trailing vortex that is possibly formed due to numerical oscillations.

As discussed in section 3, an unstable Peclet number leads to velocity oscillation during the iteration process in the execution of the Navier–Stokes solver. In certain cases, this oscillation may be not totally damped during numerical iteration, thus affecting the flow field. In physics, a vortex induced by numerical oscillation contains a considerable

amount of kinetic energy and decays rapidly through the action of fluid viscosity. According to [44], the presence of a trailing vortex further affects the airfoil, and this effect would result in the increase in drag force and the consequent reduction in lift force. Hence, as shown in Fig. 5.21, at $\alpha = 15°$, the drag coefficient calculated by the Sohrab solver is less than that calculated by the Navier–Stokes solver.

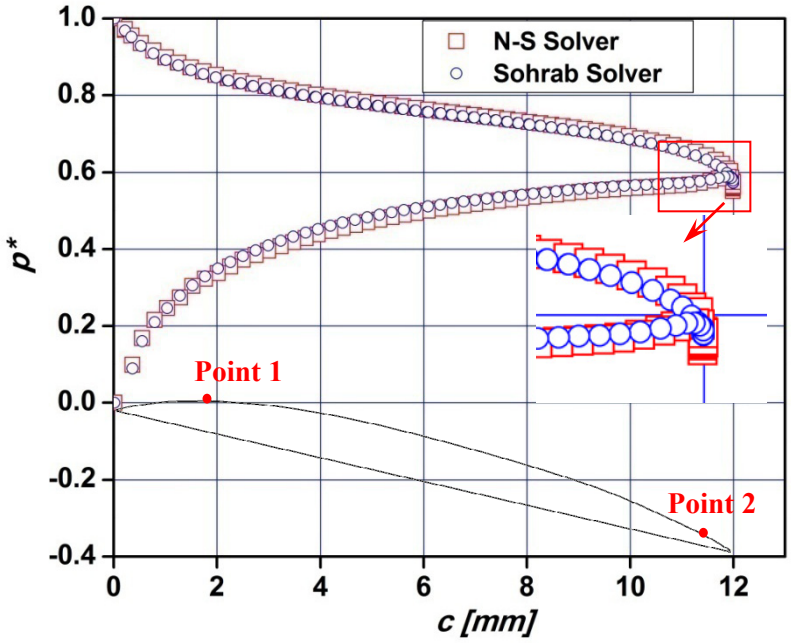


Fig. 5.24 Comparison of the pressure around the airfoil

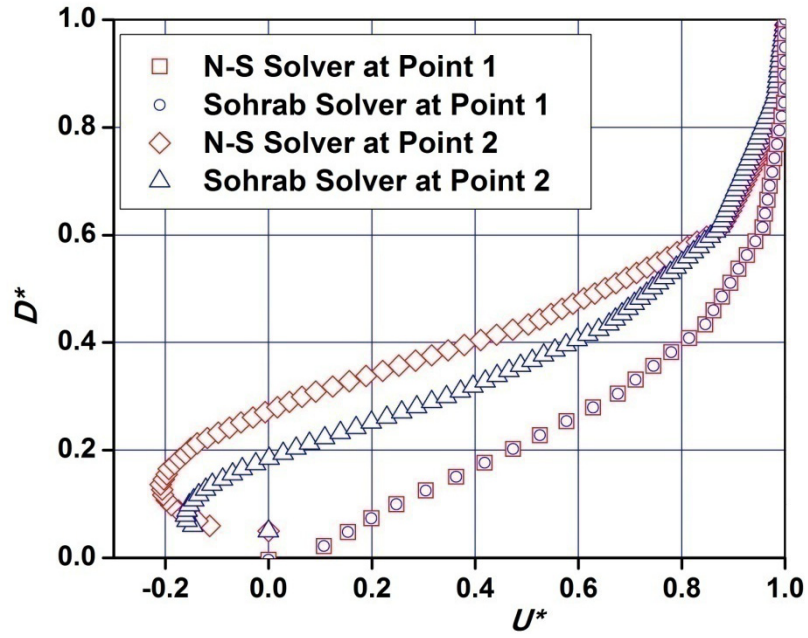


Fig. 5.25 Comparison of the velocity gradients inside the turbulent boundary layer

Fig. 5.24 shows the pressure around the airfoil. The figure clearly shows that the pressure difference calculated by the Sohrab solver is smaller than that calculated by the Navier–Stokes solver, particularly at the trailing edge of the airfoil. This result corresponds to the induction of the vortices shown in Fig. 5.22 and Fig. 5.23, thus further explaining the drag coefficient differences observed in Fig. 5.21. Further, the velocity gradients inside the boundary layer for Points 1 and 2 on the airfoil upper surface, which are illustrated in Fig. 5.24, are compared. Fig. 5.25 shows that at Point 1, which is close to the leading edge, the velocities calculated by both solvers coincide well with each other. However, at Point 2, which is close to the trailing edge, the effects of the trailing vortex result in a considerable difference in the velocity distribution. Therefore, at small angles of attack when no vortex is present in the flow field, the Sohrab solver can deliver results similar to those delivered by the Navier–Stokes solver, whereas as the angle of attack increases, the vortex begins to strongly affect both pressure and velocity fields, leading to a large difference between the solutions obtained by both solvers.

Following the calculation procedure of the Sohrab solver in the case of the airfoil model, the need to calculate the corresponding convective velocity for turbulence simulations is recognized. As shown in Fig. 2.4, numerical simulation initiates by obtaining the potential flow solution and continues with the calculation of the laminar flow. This procedure, which consumes much shorter computation time as compared with the turbulent flow computation, can be considered as the initial condition for the turbulence calculation that would follow the calculation of laminar flow. On the other hand, in the case of the Navier–Stokes solver, the same procedure is assumed to initiate turbulence calculation with a smooth initial condition in OpenFOAM [45]. Therefore, in order to compare the calculation time taken by the two solvers, the aforementioned initial process can be ignored and only the turbulence calculation process should be carried out.

5.3 Secondary Flow in a Curved Duct

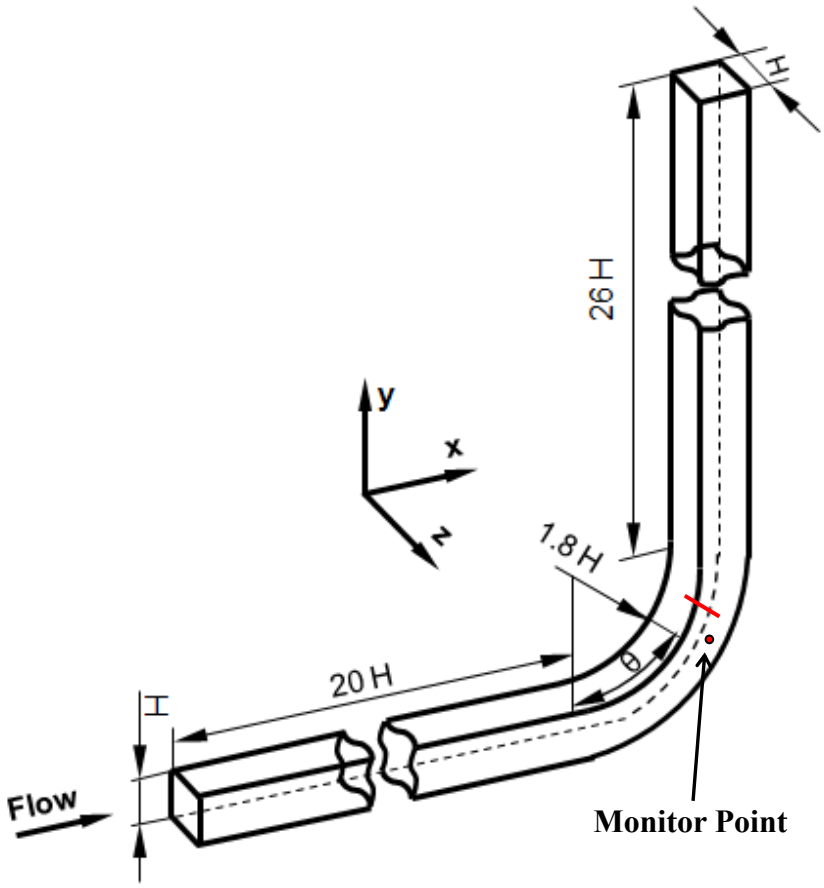


Fig. 5.26 Geometry of the curved duct

In the previous sections, two classical flow models were simulated by using the Sohrab solver, and the results were found to be in good agreement with those obtained by the Navier–Stokes solver. Such encouraging results stimulated the further applications of the Sohrab solver in engineering flow models.

Thus far, the Sohrab solver was only validated in terms of 2D flow. However, the code of the solver was developed in a 3D FVM discretisation environment. In order to assess the ability of the Sohrab solver to perform 3D calculations, turbulent secondary flow effects are considered, and consequently, a 90° curved duct is simulated with the corresponding results presented in this section.

As illustrated in Fig. 5.26, the curved duct considered for the present calculation has a square cross section and a bend radius in the middle centreline to duct height ratio of

$$\frac{R}{H} = 2.3 \tag{5.12}$$

In order to obtain a fully developed parabolic velocity distribution before the flow enters the bend, the upstream length of the duct is set to 20*H*. On the other hand, the downstream length of the duct is set to 26*H* to ensure a satisfactory boundary condition at the outlet of the duct for carrying out the numerical calculations.

5.3.1 Numerical Simulation Setup

With a Reynolds number $Re = 40{,}000$, a turbulence boundary condition is considered in the case of the curved duct. By following the aforementioned methods for calculating convective velocity, both Sohrab and Navier–Stokes solvers are applied to carry out the numerical solution for the flow field in the curved duct using the standard k–ε turbulence model. The detailed boundary conditions and numerical schemes are listed in Tab. 5.9 and Tab. 5.10.

Inlet	$\mathbf{U}_{in} = 1$ m/s, $\mathbf{w} = \mathbf{U}_{laminar}$, 5% turbulence intensity
Outlet	Average static pressure, $p = 0$ Pa
Wall	No–slip B. C.
Working fluid	$\nu = 1.0\times10^{-6}$ m^2/s
Reynolds number	$Re = 4.0\times10^{4}$, turbulent flow
Turbulence model	k–ε

Tab. 5.9 Boundary conditions setup for the flow in a curved duct

Component:	**Scheme:**	**Numerical behaviour:**
Convection	TVD	Second order, bounded
Diffusion	CD	Second order, unbounded
Pressure gradient	CD	Second order, Gaussian integration

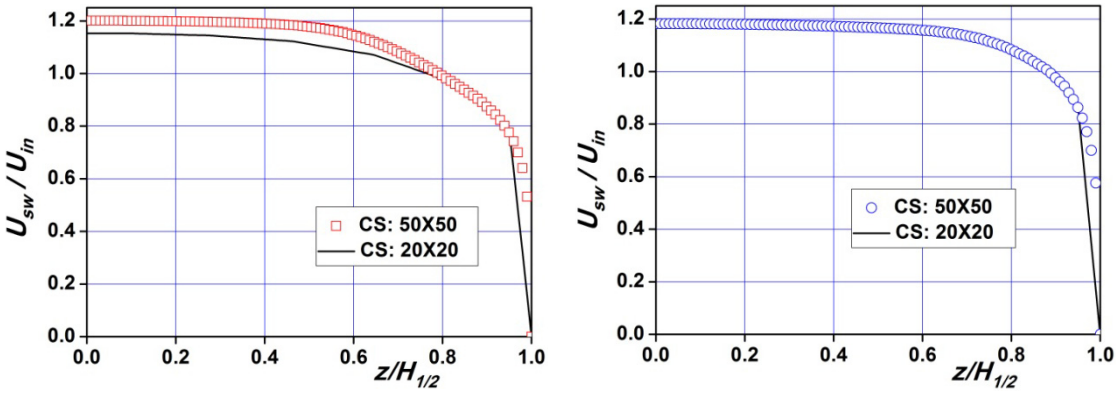
Tab. 5.10 Numerical schemes setup for the flow in a curved duct

3D flow simulation requires large computer resources because of the complexity of the 3D mesh. An insufficient mesh resolution introduces mesh–induced errors into the numerical solution. In detail, if the solution changes rapidly in some regions of the domain and the local number of computational points is not sufficient to describe that change, the shape of the solution will effectively be lost. In finite volume calculations, the accuracy of interactions described by the differential equation strongly depends on the mesh resolution. As discussed in Chapter 3, the presence of the convective velocity in the Sohrab solver makes numerical errors stable, thus reducing the requirement of the quality of the mesh for the simulation. This has been proved in section 5.1.1.4 by comparing the numerical errors of the Sohrab solver with those of

the Navier–Stokes solver, when a simple convective velocity is applied for the boundary layer flow. It is therefore important to investigate the numerical situations when the complicated convective velocity is applied in the Sohrab solver to solve the modified equation of fluid motion. Due to this reason, the requirement of the mesh quality for the two solvers is discussed for the present 3D curved duct model.

By varying the resolution at the cross section of the duct, two structured meshes are applied for the two solvers for tests:

1) 400 (20 × 20) nodes for the cross section,
2) 2,500 (50 × 50) nodes for the cross section.



Navier–Stokes solver Sohrab Solver

Fig. 5.27 Mesh effect on velocity profiles

Using the aforementioned meshes, both Sohrab and Navier–Stokes solvers calculate the turbulent secondary flow in the bending. As shown in Fig. 5.27, the velocity profiles in streamwise direction inside the bending at the red line in Fig. 5.26 are plotted along z–direction. Near the wall ($z/H_{1/2}$ = 1.0) a big difference is presented in both figures, because the low quality mesh results in a large y^+ value in the wall function. Except this region, it is clear that, the Navier–Stokes solver produces a 10% difference in the results for the different mesh models, while the Sohrab solver produces the same accuracy for both mesh models. This result supports the analysis in Chapter 3 that the Sohrab solver can reduce the numerical error caused by the mesh resolution, which is a significant advantage of the solver in CFD applications. However, in order to give an equivalent comparison for the computation time, the high quality mesh is applied to both solvers for the further simulations. Since the 3D curved duct case stands for the maximum mesh cost simulation, the mesh quality induced error will not again be discussed for the other flow models. Then, as shown in Fig. 5.28, the mesh is fixed to 50 × 50 for the square cross section, resulting in a total grid number of 437,500. Close to the wall, a finely graded mesh is plotted and the first grid node near the wall is well controlled to reach a satisfactory y^+ value of $y^+ < 30$.

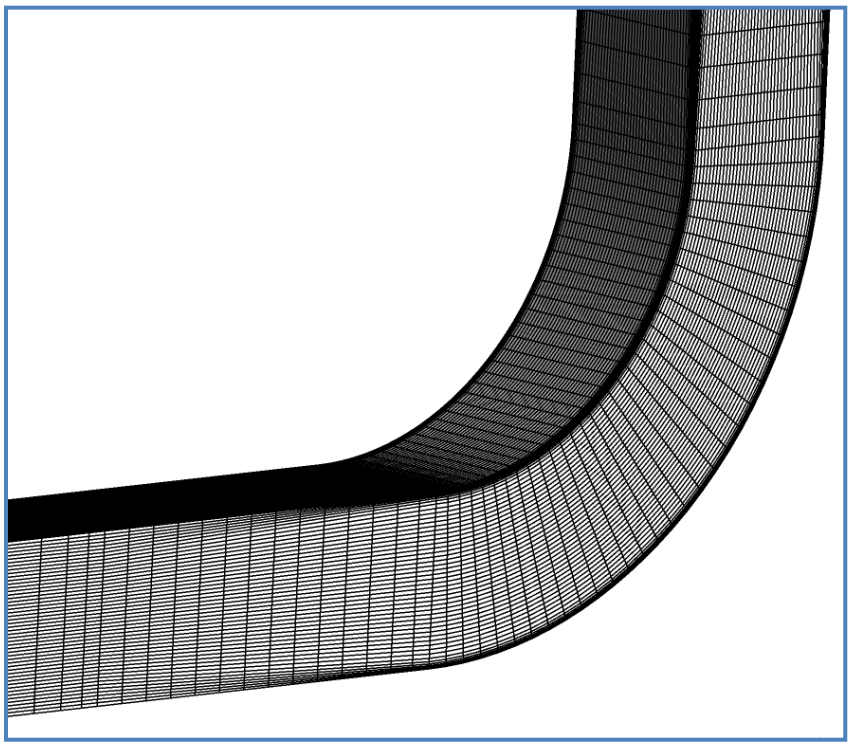


Fig. 5.28 Section of the grid of the curved duct

5.3.2 Results & Comparisons

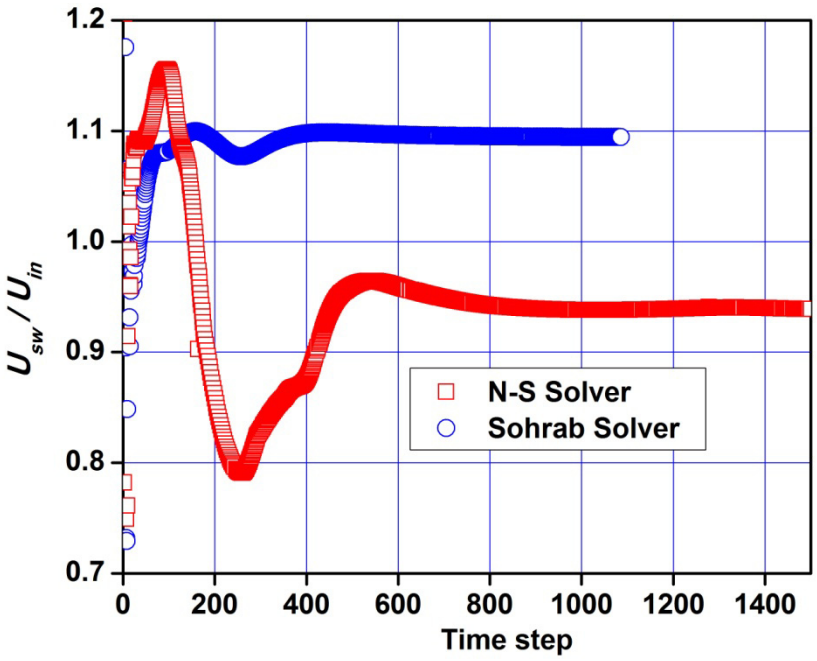


Fig. 5.29 Comparison of the convergence process of the velocity

After assessing the velocity fluctuation at a pre–defined point (monitor point), which is shown in Fig. 5.26, computational convergence procedures for both solvers are carried

out, and the results are compared. As shown in Fig. 5.29, the two velocities experience numerical oscillations and converge in the given time steps. In terms of the convergence procedure, a considerably larger velocity oscillation is observed in the Navier–Stokes solution as compared to the Sohrab solution. Such oscillations increase the instability in numerical calculations, which in turn tend to fail. Furthermore, the damping of such oscillation requires additional time steps, thus increasing computation time. As shown in Fig. 5.29, the Sohrab solver uses less than 500 time steps to achieve convergence while the Navier–Stokes solver requires more than 1,000 time steps, which are twice that required by the Sohrab solver. This can be explained by the numerical stability analysis carried out in Chapter 3. The stable Peclet number and numerical diffusion calculated by the Sohrab solver dampens the numerical oscillation and allows the calculation to produce solutions that converge at a high speed. However, the two solvers produce different velocities when convergence of their solutions is reached. In the following figures, this difference in the velocities will be compared to the measured data.

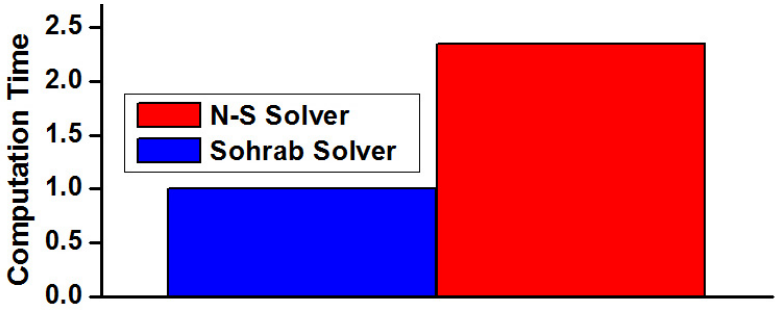


Fig. 5.30 Comparison of computation time taken by both solvers

Fig. 5.30 shows a comparison of the computation time taken by the two solvers. The Navier–Stokes solver is twice slower than the Sohrab solver in completing the simulation. This result coincides with the analysis discussed above and shows the linear advantage of the Sohrab solver to solve a 3D flow problem.

In order to study the development of the secondary flow motion, the velocity vectors are plotted at three cross sectional planes. As shown in Fig. 5.31, both Sohrab and Navier–Stokes solvers produce more or less similar secondary flow patterns. Upstream the bend, at $x/H = 0.25$, the secondary flow effect is so weak that no vortex is presented and the fluid is just displaced towards the inside of the duct. In the bend, at $\theta = 60°$, the development of a pair of rotating symmetric secondary vortices is induced by the curvature of the duct. Initially, the vortices carry near–wall fluid to the inside of the bend and return the low momentum fluid near the centre from the inside towards the outside of the bend. However near the boundary layer along the outside wall, the low streamwise velocity cannot continue to develop the centrifugal force against the radial

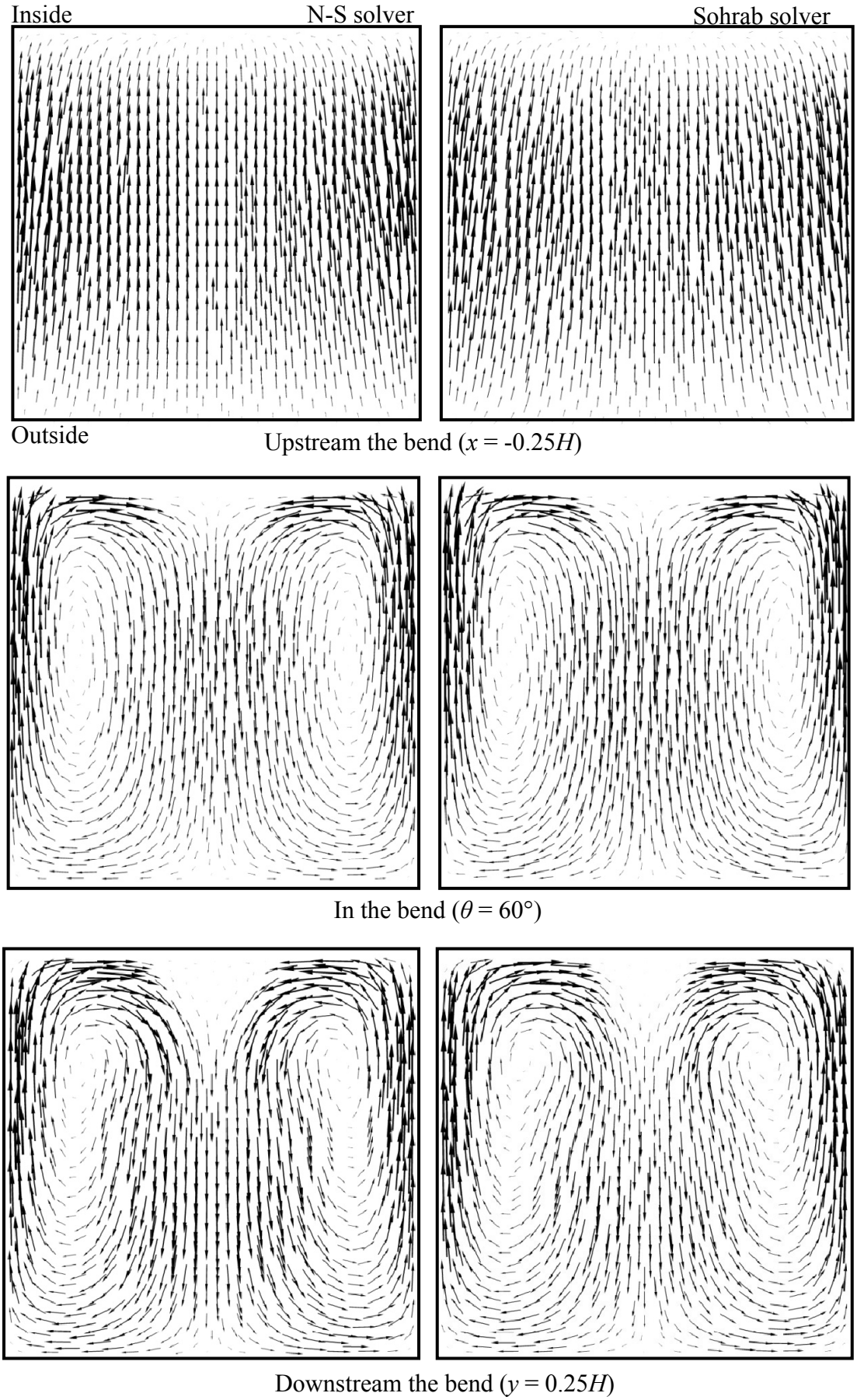


Fig. 5.31 Vector plots of the velocity at various cross sections

pressure gradient, therefore the fluid is turned back towards the inside wall. Downstream the bend, at $y/H = 0.25$, the vortices continue to develop and their cores move towards the inside of the bend.

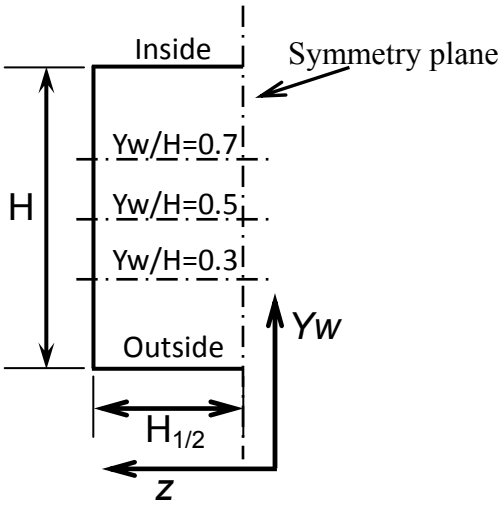


Fig. 5.32 Sketch of the cross section of the curved duct

Then, the streamwise and crosswise velocities are compared quantitatively at the two locations where the vortices can be found: in the bend (at $\theta = 60°$) and downstream the bend (at $y/H = 0.25$), with different Yw positions, as sketched in Fig. 5.32. At $\theta = 60°$, the crosswise and streamwise velocities calculated by both solvers are compared with the measured data points that were reported in [46]. In the case of crosswise velocity shown in Fig. 5.33, secondary flow features can be observed. The fluid near the symmetry plane moves away, whereas that close to the wall moves in the opposite direction. Both numerical solvers reproduce such a phenomenon. As shown in Fig. 5.34, the streamwise velocity profiles obtained from the numerical calculations are in a good agreement with the measurements. It is observed that near the wall of the duct, the Sohrab solver performs better to deliver results that coincide with the measured data. In terms of streamwise velocity, the results produced by the Sohrab solver agree well with the measured data points except for a few points in principle. Downstream the bend at $y/H = 0.25$, the secondary flow feature continues to develop and the resultant variation in the velocity is assumed to be greater downstream the bend than inside it. The numerical results are compared with the measured data [46]. In the crosswise direction as shown in Fig. 5.35, the Navier–Stokes solver shows a very high velocity variation, particularly near the wall of the duct. This observation proves that the strength of the secondary flow calculated by the Navier–Stokes solver is greater than the measured strength of the secondary flow. In the case of the Sohrab solver, the secondary flow calculated near the wall sufficiently agrees with the measured data, whereas the secondary flow calculated near the symmetry plane is observed to be low. For streamwise velocity profiles as shown in Fig. 5.36, it can be confirmed that both solvers can predict secondary flow development very well.

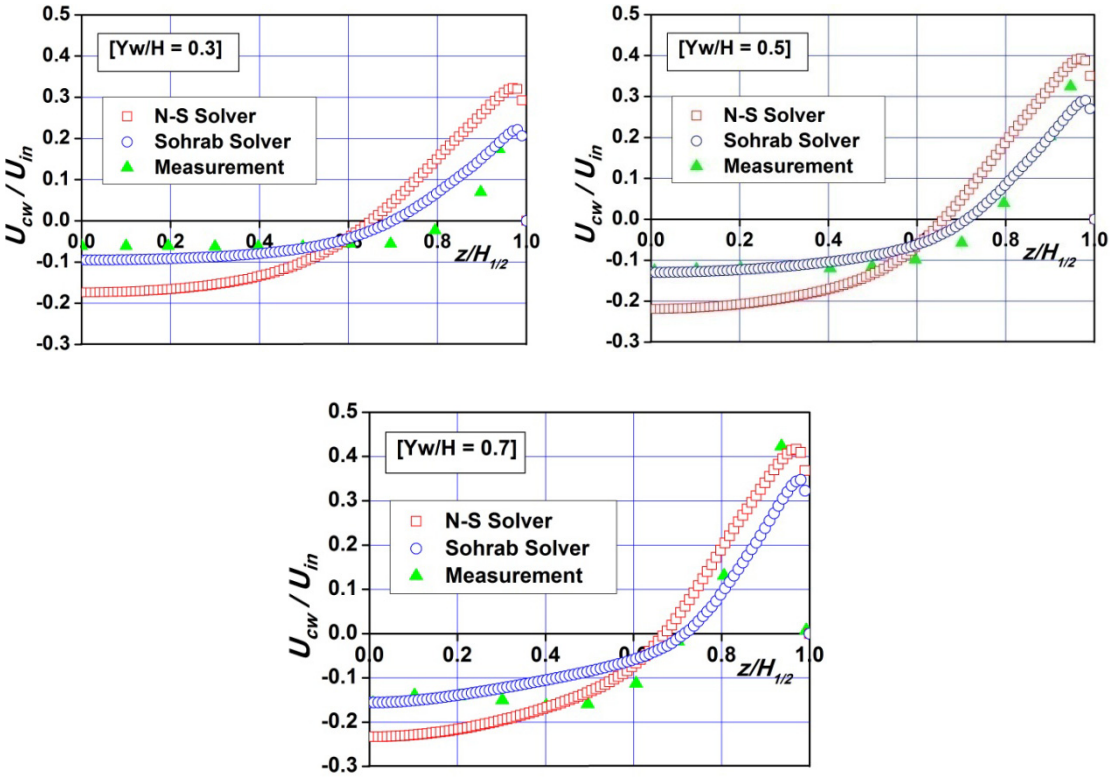


Fig. 5.33 Comparison of the crosswise velocity profiles in the bend at $\theta = 60°$

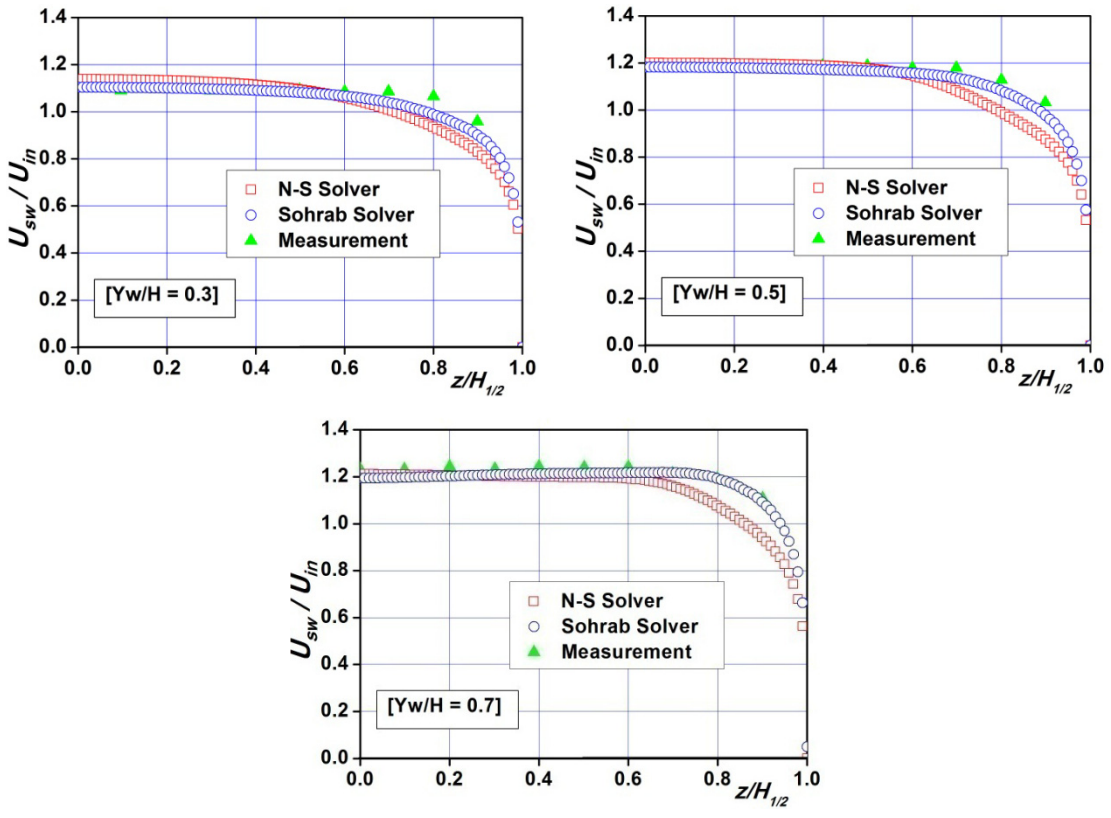


Fig. 5.34 Comparison of the streamwise velocity profiles in the bend at $\theta = 60°$

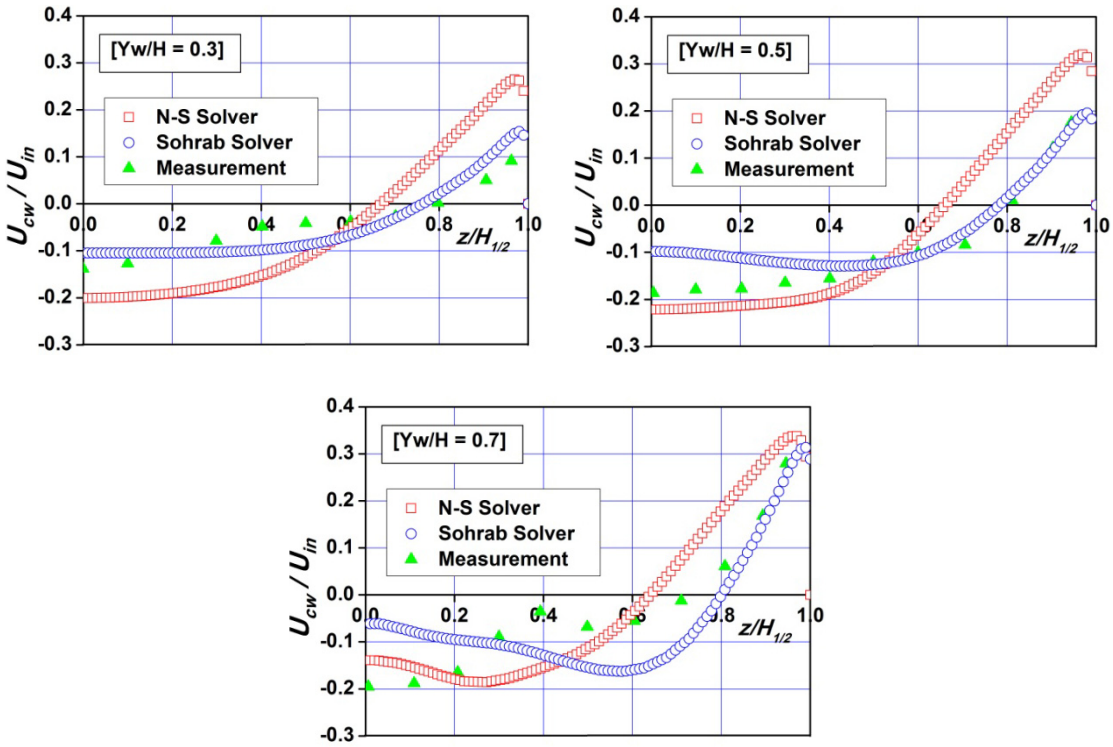


Fig. 5.35 Comparison of the crosswise velocity profiles downstream the bend

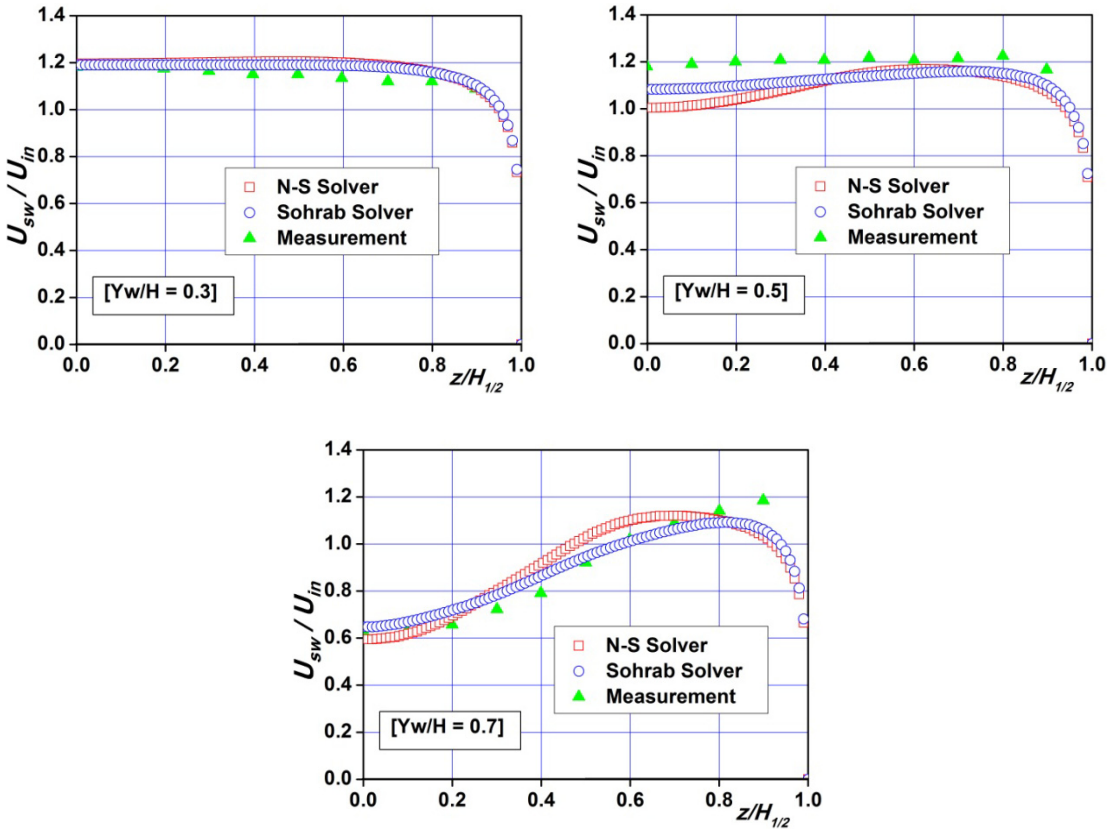


Fig. 5.36 Comparison of the streamwise velocity profiles downstream the bend

5.4 Secondary Flow in Cascades

In the sections above, the physical boundary conditions setups for the Sohrab solver including the convective velocity determining process and the numerical issues have been discussed in detail to recover its linear advantages over the Navier–Stokes solver. Now the applications of the Sohrab solver in turbomachinery flow models are under consideration. The final solution for this aim has to face transient 3D flow of viscous fluid through rows of closely spaced blades. Considering the complexity of the problem, it seems likely that complete relationships have to be established step by step. Therefore the solver application started with the flow analysis in steady state, thus neglecting effects of time dependence and mesh movement. Such a configuration of the numerical solver is accepted for the cascade simulation. Tests of a cascade can represent many fundamental data concerning the performance of turbomachines under a wide range of operation conditions with relative ease, rapidity and low cost.

In this section, a 2D and a 3D cascades are simulated by the Sohrab solver under turbulent flow environment. By following the experiment of Herrig et al. [47], a 2D cascade for axial–flow compressor is solved first. On the basis of the experience of the 2D calculation, furthermore, a 3D annular cascade chosen from the experiment of Wang [48] is simulated. The Sohrab solver results and comparisons with those obtained from the Navier–Stokes solver and experiment [47], [48] are presented in this section.

5.4.1 2D Cascade

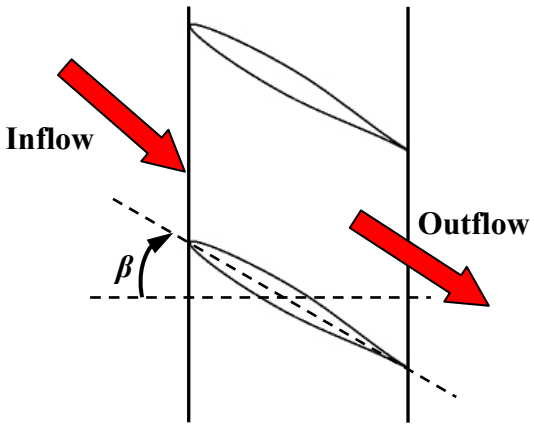


Fig. 5.37 Sketch of the 2D cascade

Starting with a relative easy flow condition, first the 2D turbulent flow though a cascade is under consideration. The model used in the simulation is the Langley five–inch low speed, seven–blade cascade. A 2D sketch of this cascade is shown in Fig. 5.37. This cascade is obtained from the experiment of Herrig et al. [47]. The 2D geometric property of the cascade insures the two dimensionality of the flow. The

airfoil profile of the blade was formed by combining a basic thickness form which is based on NACA 65(216)–010 with cambered mean line. The experiment was processed at a flow speed considered incompressible, thus resulting in a blade chord Reynolds number Re = 245,000. Uniform upstream flow was preserved by the proper test conditions.

5.4.1.1 Numerical Simulation Setup

In the numerical simulation, only one blade is considered with the translational cyclic boundary condition to simulate the 2D flow through the cascade. In detail, it is calculated with the inlet angle $\beta = 30°$ over angle of attack α of 3°, 11° and 17°.

The same mesh is applied to both Sohrab and Navier–Stokes solvers to allow a direct comparison of the results. The grids of the part of the calculation domain are shown in Fig. 5.38. In order to present a better view, the grids near the leading and trailing edges are enlarged in Fig. 5.39. The first grid node near the airfoil is well controlled to preserve y^+ below 5, which offers a perfect condition for the wall function in OpenFOAM to resolve the boundary layer of the turbulent flow.

2D turbulent flow is computed by solving the Reynolds averaged equations with k–ε turbulence model either for the Navier–Stokes solver or Sohrab solver. For boundary conditions, which are summarized in Tab. 5.11, the flow velocity is imposed at the inlet while the static pressure is considered at the outlet. A smooth and no–slip wall condition is assigned for the flow at the blade. Considering the analysis in section 5.3, a laminar flow field has to be solved as the initial boundary condition for the following turbulence simulation. For numerical schemes, the setup in Tab. 5.10 is used to execute the spatial discretisation.

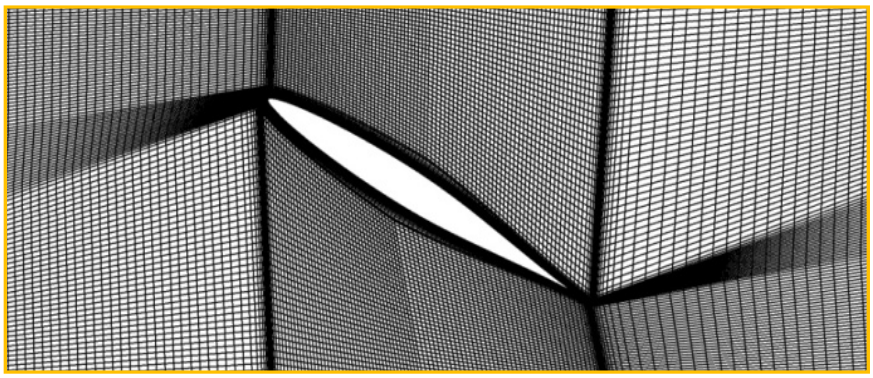


Fig. 5.38 Part grid region of 2D cascade

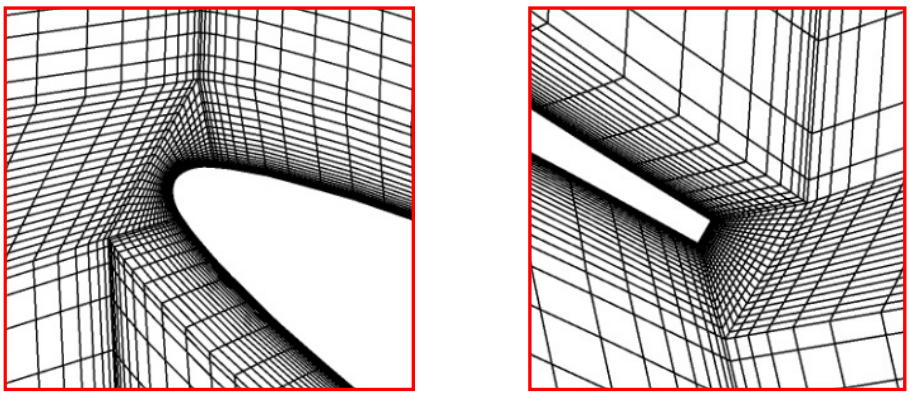


Fig. 5.39 Grids near leading edge and trailing edge

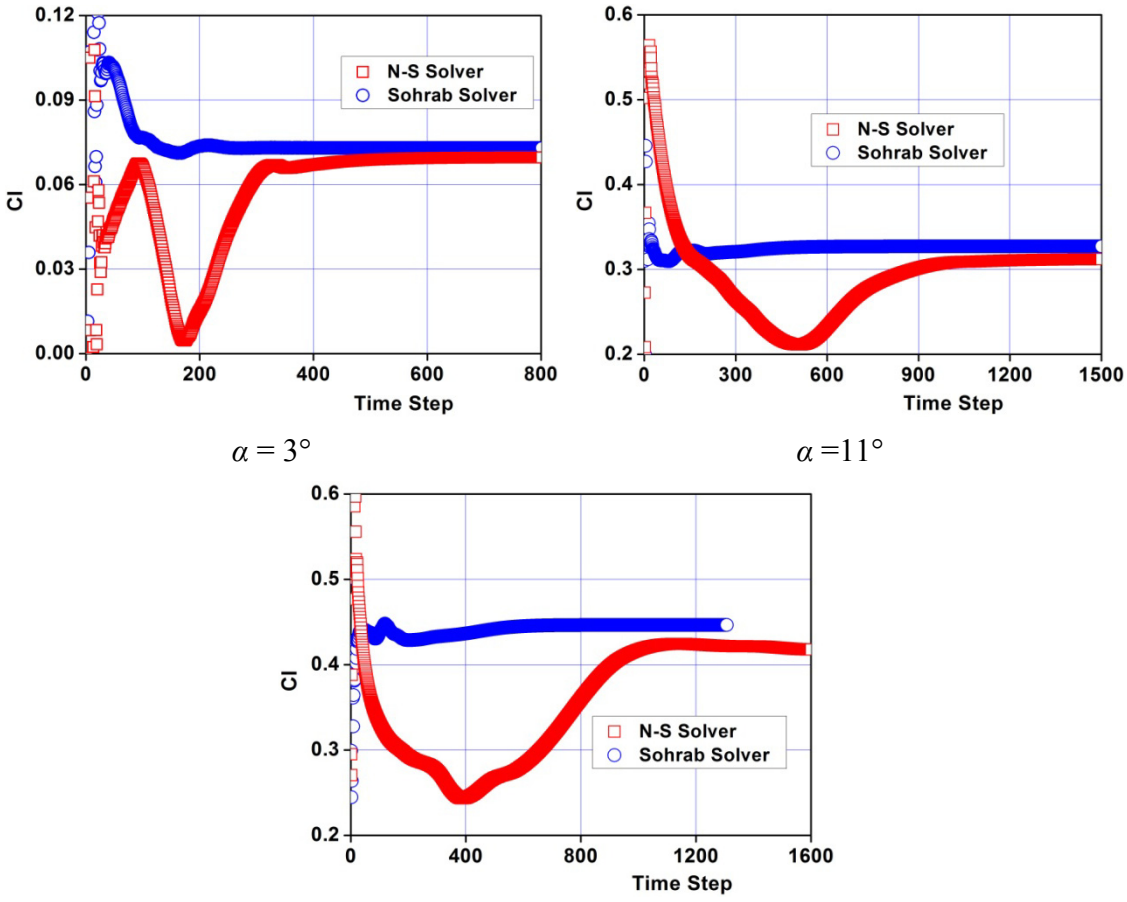
Inlet	$\mathbf{U}_{in}$, $\mathbf{w} = \mathbf{U}_{laminar}$, 5% turbulence intensity
Outlet	Average static pressure, $p = 0$ Pa
Blade	No–slip B. C.
Symmetric walls	Symmetric B. C.
Cyclic patches	Translational periodic B. C.
Working fluid	$\nu = 1.5\times10^{-5}$ m^2/s
Reynolds number	$Re = 2.45\times10^{5}$, turbulent flow
Turbulence model	k–ε

Tab. 5.11 Boundary conditions setup for the flow in a 2D cascade

5.4.1.2 Results & Comparisons

The computational convergence procedure for both Sohrab and Navier–Stokes solvers are compared. The lift coefficient (Cl) calculation processes over the time step for three angles of attack are shown in Fig. 5.40 respectively. The lift coefficient defined in Eq. (5.13) is a dimensionless coefficient that relates the lift force generated by a lifting body, the dynamic pressure of the fluid flow around the body, and a reference area associated with the body.

$$Cl = \frac{2\mathbf{F}_L}{\rho c l \mathbf{U}^2} \tag{5.13}$$



$\alpha = 3°$ $\alpha = 11°$

$\alpha = 17°$

Fig. 5.40 Comparison of the convergence processes of the lift coefficient

At each angle of attack, the comparison of the lift coefficient convergence processes obtained by the two solvers proves that the Navier–Stokes solution has to experience larger numerical oscillation as compared to the Sohrab solution. This result coincides with the numerical error analysis in Chapter 3 and is similar to the velocity convergence process presented in section 5.3.2.

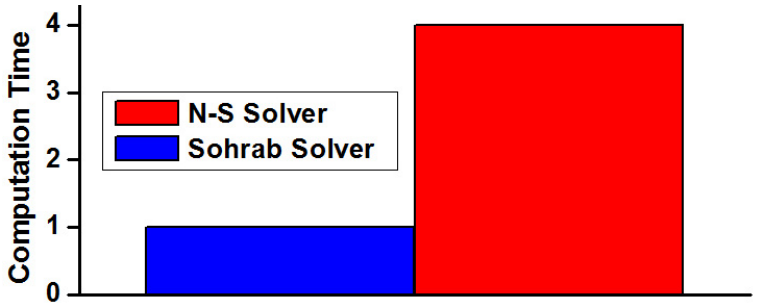


Fig. 5.41 Comparison of the averaged computation time taken by both solvers

When the two solvers solutions converge, the computation time taken by each case is compared and then an averaged computation time ratio is plotted in Fig. 5.41. The

computation time taken by the Navier–Stokes solver is about four times of that taken by the Sohrab solver. This is because the Navier–Stokes solver not only requires more time steps to reach the convergence of the iterations, but also needs more time than the Sohrab solver to complete the computation in each time step as discussed in Chapter 4.

The results obtained by the numerical solvers are compared with the measured data from [47] to evaluate the calculation accuracy of the Sohrab solver. The pressure coefficient (*Cp*), which is defined in Eq. (5.14), is selected for the result evaluation.

$$Cp = \frac{2(p - p_\infty)}{\rho U_\infty^{\,2}} \tag{5.14}$$

Fig. 5.42 presents the pressure coefficients on the blade surface calculated by the numerical solvers, along with the published data measured by Herrig et al. [47]. Generally, the Sohrab solver results agree with that of the Navier–Stokes solver and the measured data for all the three angles of attack, thus providing a validation for the Sohrab solver in turbomachinery flow applications. Evidently, the pressure difference between the upper and lower surfaces of the blade is strongly dependent on the angle of attack, growing with higher angle of attack.

In terms of the pressure difference between the upper and lower surfaces of the blade, the numerical solutions always show larger difference than the measurements, especially near the leading edge, which can be found in the results of all three angles of attack. At $\alpha = 3^o$, a noticeable difference near both leading and trailing edges can be found between the numerical results and the measured data. Near the leading edge, the numerical solutions present a larger pressure difference than the measurement, however, near the trailing edge, the former show a smaller pressure difference than the later. The measured data near the trailing edge may be disturbed during the experiment because it is greatly different to not only the numerical results, but also the measured data at another two angles of attack. Another important phenomenon is that near the trailing edge, the Sohrab solver delivers smaller pressure coefficient than the Navier–Stokes. This result can also be observed in the solutions of the other two angles of attack. At $\alpha = 11^o$, the numerical solutions do differ slightly from the measurement. The two types of results match each other better than the other two angles of attack. At $\alpha = 17^o$, obvious difference can be found near the trailing edge of the blade. In the experimental results, the pressure measured at the upper surface is nearly the same as the one at the lower surface, while in the numerical solutions, a difference between the pressures at the two surfaces can be found.

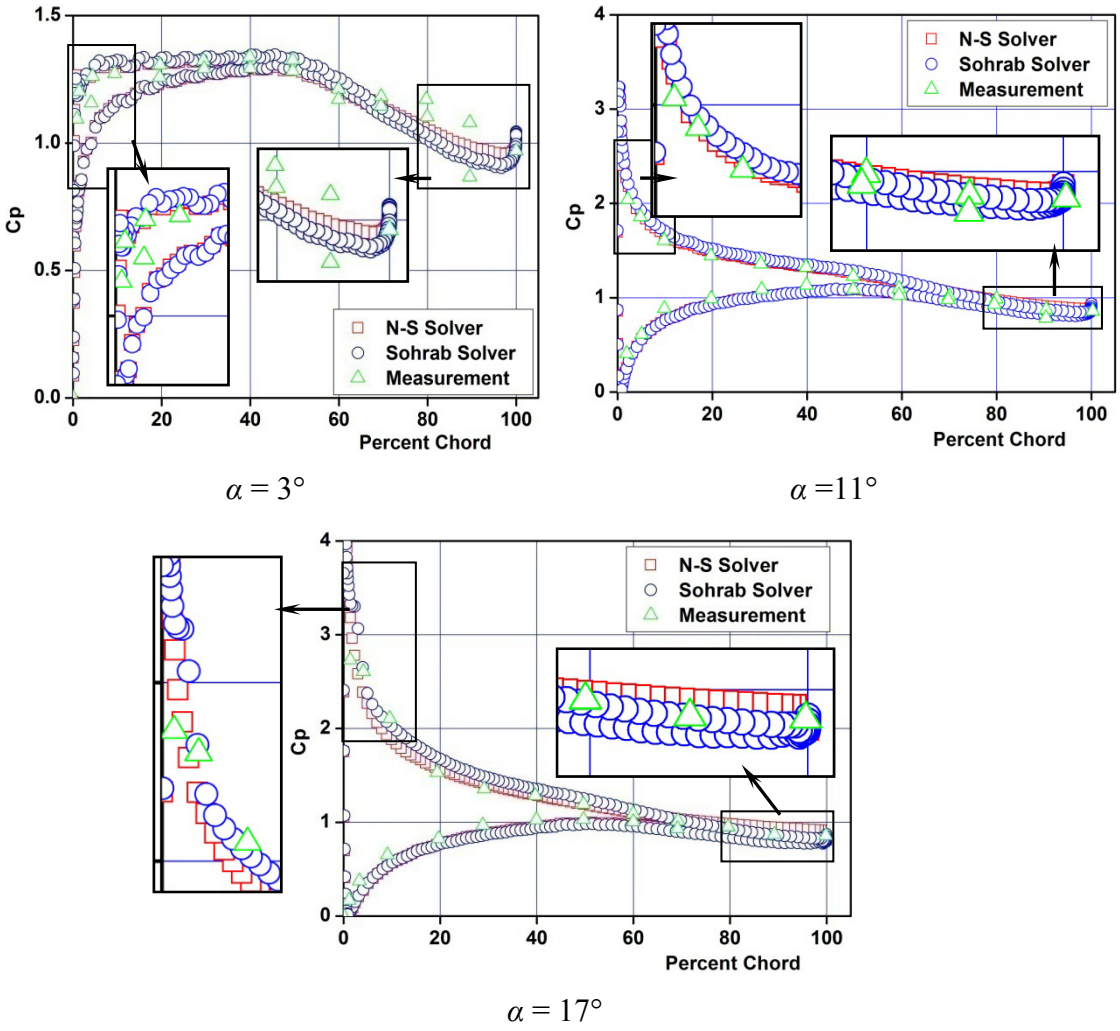


Fig. 5.42 Comparison of the pressure coefficient

The lift coefficient is formed by the pressure difference. Therefore the comparison of this coefficient will give another proof for the above pressure distribution analysis. As illustrated in Fig. 5.43, the results obtained by the numerical solutions have a satisfactory agreement with the measured data, especially at low angles of attack. This result coincides with the pressure distribution analysis. In detail, a 10% difference between the Sohrab solver and measurement results can be found at $\alpha = 17°$. The reason for this difference plausibly arises from the wing span. In the measurement, the cascade is tested with a finite wing span, while 2D cascade with infinite wing span is applied to the numerical simulations. According to Clancy [49], on a wing of finite span, the pressure difference between the upper and lower surfaces causes the air to flow from the lower surface root, around the wing tip towards the upper surface wing root. This span–wise flow of air combines with chord–wise flowing air, causing a change in space and direction, thus providing vortices around the airfoil trailing edge. This phenomenon can be found in Fig. 5.42. At $\alpha = 17°$ the pressure distribution near the trailing edge shows obvious difference between the numerical results and the measured data. The vortices are unstable, and they quickly combine to produce wing tip vortices. The

resultant vortices decrease the effectiveness of the wing to generate lift force. Hence, as compared to the numerical simulation which deals with an infinite airfoil, the measurement shows a smaller lift coefficient at high angle of attack in the result. Since the Navier–Stokes solver always produces smaller pressure difference, at $\alpha = 17°$, it shows a better agreement with the measurement than the Sohrab solver.

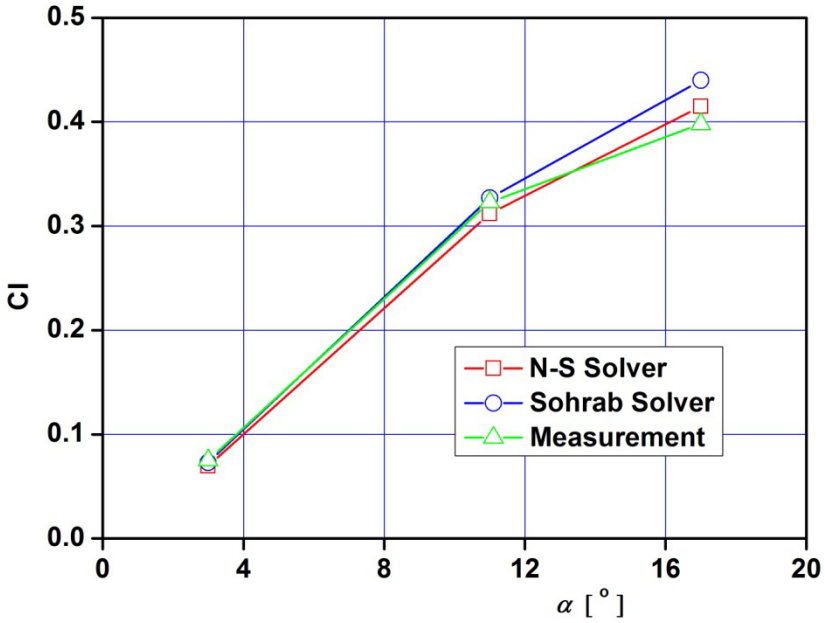


Fig. 5.43 Comparison of the lift coefficients

5.4.2 3D Cascade

As compared to the 2D cascade, strong 3D secondary flow can be found in the high turning angle turbine blade cascade. Such a flow is considered to greatly deteriorate the performance characteristics of blades. Therefore, it is important for the Sohrab solver to predict the complicated behavior of the internal flow in the 3D cascade passage appropriately.

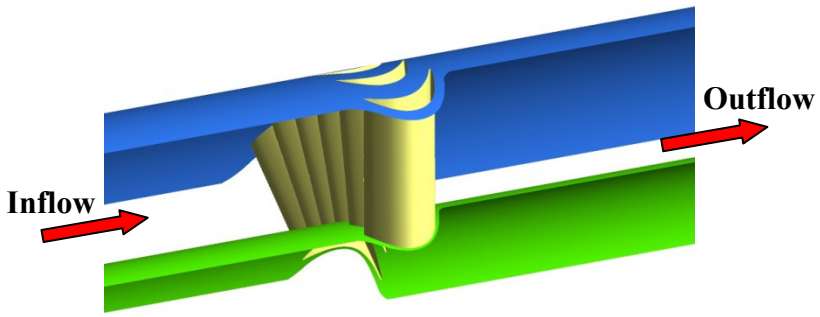


Fig. 5.44 3D model of the six–blade annular cascade

As shown in Fig. 5.44, the flow model of 3D turbulent flow in a low–speed annular cascade is considered for the numerical simulation. The cascade model is chosen from the experiment of Wang [48]. The test rig was designed as an annular sector row mounting six high turning angle turbine blades circumferentially. Uniform inlet flow direction was preserved to provide a sufficient flow condition for the measurement. More detailed information about the experimental facilities can be found in [48].

5.4.2.1 Numerical Simulation Setup

For the CFD computations, both Sohrab and Navier–Stokes solvers are employed on a 3D structured mesh to simulate the turbulent flow in the annular cascade. A section view of the grids is shown in Fig. 5.45. In addition, the details near the leading and trailing edges are enlarged in Fig. 5.46. In order to offer a satisfactory boundary layer solution for the blade, y^+ is controlled below 60 for the cascade domain. The total grid number is about 200,000.

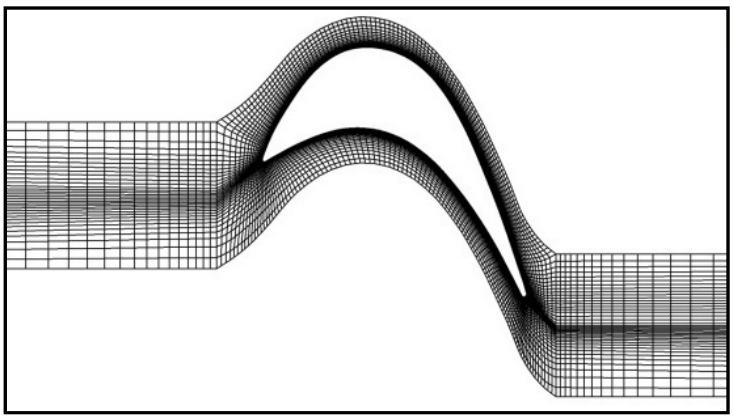


Fig. 5.45 Part grid of annular cascade

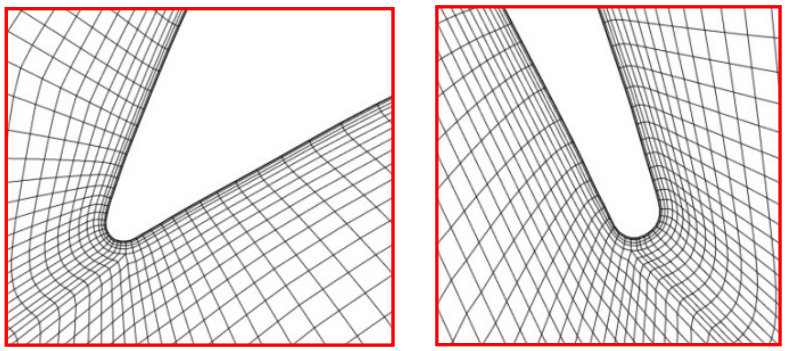


Fig. 5.46 Grids near the leading and trailing edges

Assuming periodic flow inside the cascade, only one blade flow region is simulated by using the rotating cyclic condition in OpenFOAM. The simulation boundary conditions which are listed in Tab. 5.12 are similar to those used for the 2D cascade simulation.

For numerical schemes, the setup in Tab. 5.10 is applied.

Inlet	$\mathbf{U}_{in}$ = 10.81 m/s, $\mathbf{w} = \mathbf{U}_{laminar}$, 5% turbulence intensity
Outlet	Average static pressure, p = 0 Pa
Airfoil	No–slip B. C.
Symmetric walls	Symmetric B. C.
Working fluid	$\nu = 1.34\times10^{-6}$ m^2/s
Reynolds number	$Re = 1.0\times10^{5}$, turbulent flow
Turbulence model	k–ε

Tab. 5.12 Boundary conditions setup for the flow in an annular cascade

.

5.4.2.2 Results & Comparisons

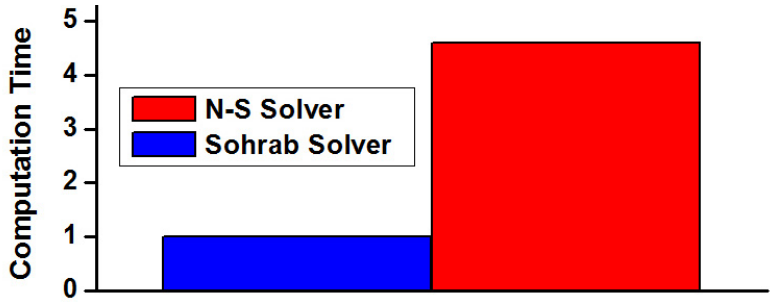
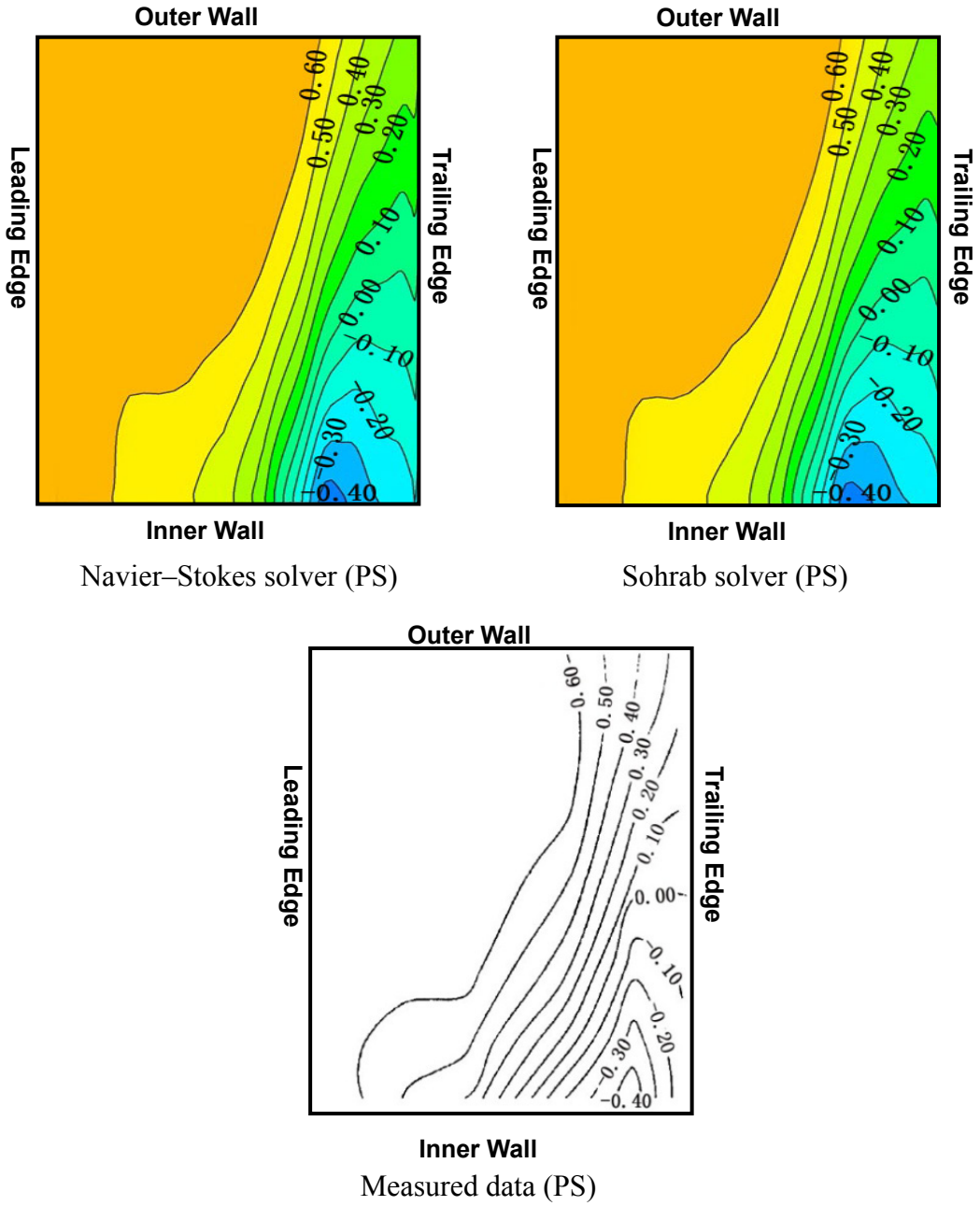


Fig. 5.47 Comparison of the computation time taken by both solvers

The computation time taken by the two solvers is compared in Fig. 5.47. As expected, the Sohrab solver shows its linear advantage in efficiency. It is more than four times faster than the Navier–Stokes solver in completing the simulation.

The static pressure contour at both pressure and suction sides of the blade obtained by the two solvers are compared with the measured data reported in [48] in Fig. 5.48 and Fig. 5.49. The comparison in these two figures shows that the numerical results match each other perfectly and agree well with the measured data.

In general, the pressure variation in radial direction is considerable smaller than that in the axial chord direction at both blade sides, because when the flow is turned by the blade row, the flow far away from the end wall can be considered as 2D flow. Near the inner wall, boundary layer flow has span–wise velocity gradients. When this flow is turned towards the trailing edge, transverse velocity components will be produced. Correspondingly, in that region a minimum pressure vortex core can be found at both sides of the blade. This secondary flow effect near the end wall indicates the 3D property of the flow in the annular cascade.

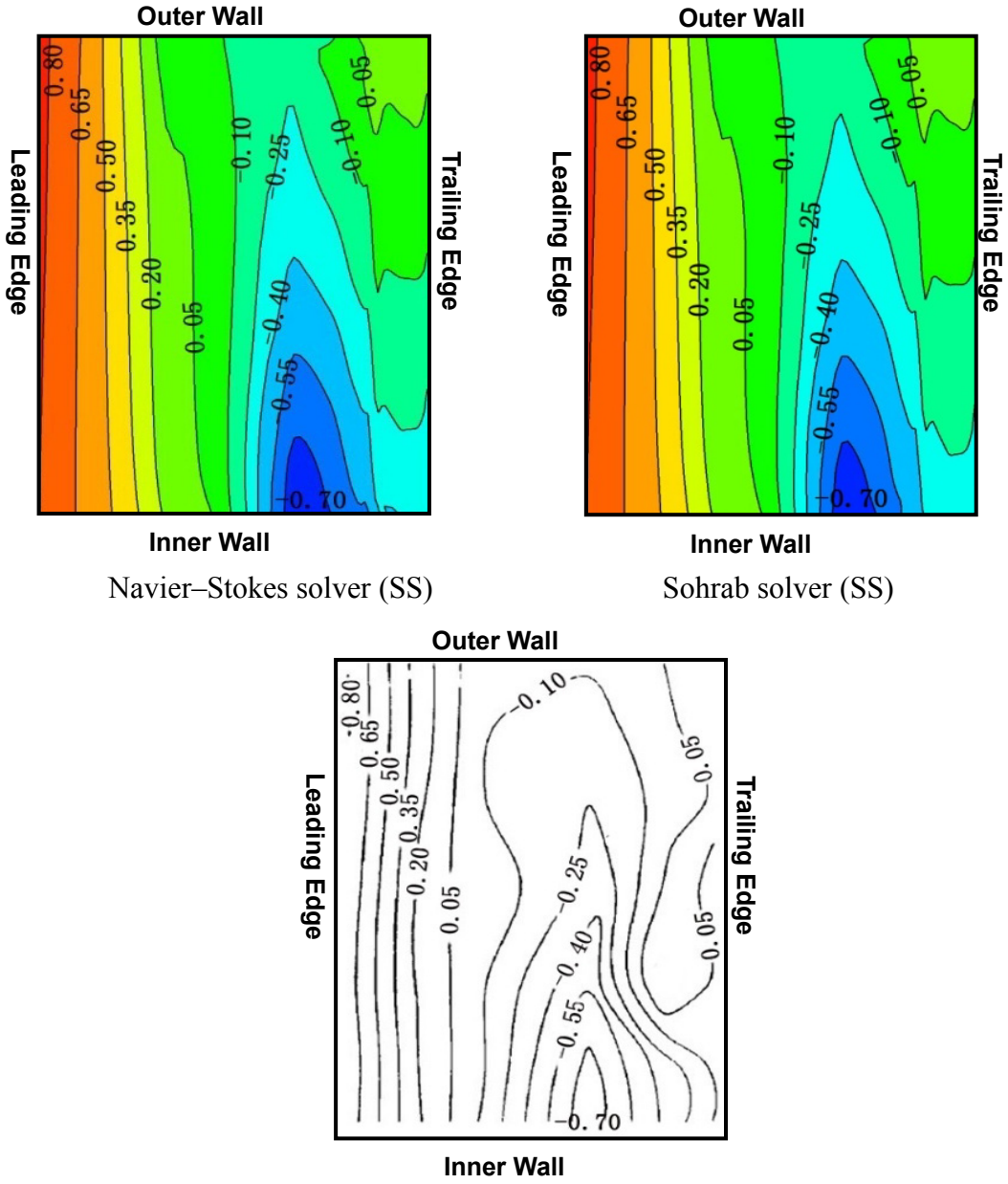


Navier–Stokes solver (PS)

Sohrab solver (PS)

Measured data (PS)

Fig. 5.48 Comparison of the pressure contours at pressure side of the blade



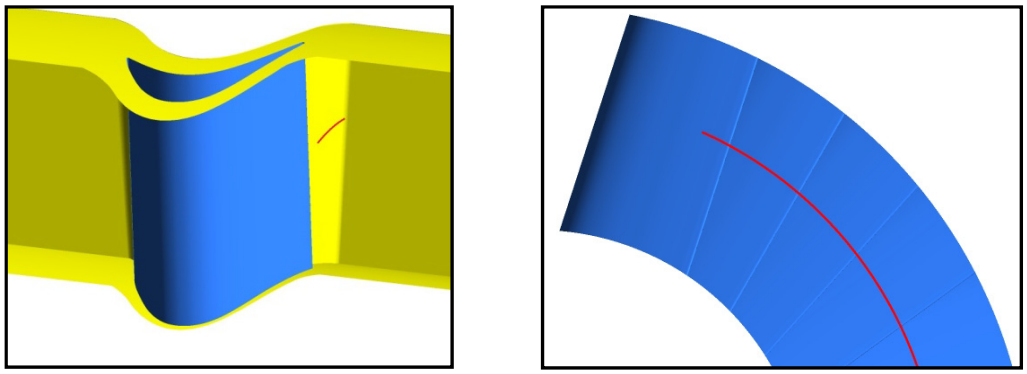
Navier–Stokes solver (SS)

Sohrab solver (SS)

Measured data (SS)

Fig. 5.49 Comparison of the pressure contours at suction side of the blade

The investigation of wake characteristics downstream the blade plays an important role in the turbomachine design process. As shown in Fig. 5.50, downstream the blade, far from the boundary walls, a polyline is plotted to illustrate the axial, circumferential and radial velocity profiles of the cascade wake to give further results comparisons.



3D view　　Section view

Fig. 5.50　Position of the polyline for the wake plot

As shown in Fig. 5.51, the velocities in three directions are plotted over the magnitude of the polyline which is represented by symbol S. As compared with the measurement, in tangential and radial directions, both numerical results show a satisfactory agreement with the measured data. However, in axial direction the numerical results present smaller variation than the measured data. This result coincides with the pressure analysis in Fig. 5.48 and Fig. 5.49. In axial direction, strong pressure gradient is presented, which usually cannot be precisely predicted by the numerical methods.

For small velocity oscillations in axial and radial directions, the numerical results calculated by the two solvers coincide well with each other. When the velocity oscillates strongly in the tangential direction, a phase shift can be observed between the two numerical results, especially in the middle portion of the results. In terms of the measured data, the Sohrab solver results are confirmed to be closer to the data points than the Navier–Stokes solver results. This is plausible because the Sohrab solver, as compared to the Navier–Stokes solver, reduced the numerical errors in the numerical solution, thus increasing the accuracy of the results for complex flow boundary conditions in turbomachinery applications.

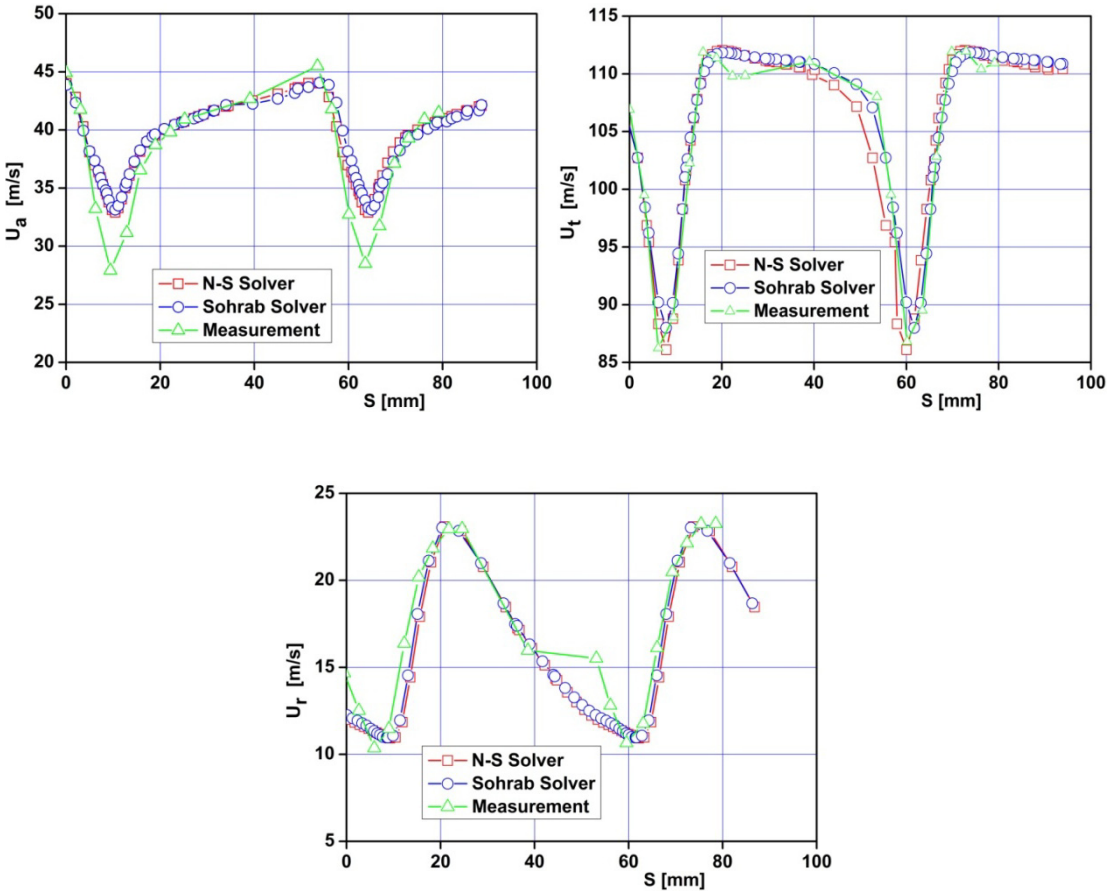


Fig. 5.51 Comparison of the wake characteristics

5.5 Rotor–Stator Interaction

The previous work offers considerable experiences and proves for the application of the Sohrab solver in CFD. They support the further application of the Sohrab solver for the numerical simulation of the incompressible flow in turbomachines, e. g. flow in a radial pump.

The intrinsically unsteady interaction between the rotating impeller and stationary diffuser which is known as rotor–stator interaction has been investigated experimentally and numerically by a number of engineers. The rotor–stator interaction can be characterized by two different mechanisms: potential flow interaction and wake interaction [50]. The potential flow interaction, which is a purely inviscid interaction between the rotor and stator blade rows moving relative to each other, results from the circulation and potential fields about the blades [51]. The wake interaction refers to the unsteadiness generated by the vertical and entropic wakes shed by upstream blade rows. These wakes are convected by the downstream blades moving relative to the wakes. Both potential flow interaction and wake interaction contribute to the development of the unsteady forces on blades both upstream and downstream. These unsteady forces not only result in hydrodynamic forces, vibration and noise, but also lead to unfavorable characteristics to pump performance at the design point [52]. As a consequence, it is important to understand and control the governing physical mechanism of rotor–stator interaction.

In the numerical investigation of the rotor–stator interaction, the physical domain of rotor rotates in time. Since this moving domain experiences large rotation, it becomes necessary to apply the dynamic mesh solver to execute the numerical simulation. In this section, 2D numerical simulations of the flow at midspans of two radial pumps are performed using both Sohrab and Navier–Stokes solvers. The results obtained by the two solvers are compared with measured data reported in experimental literatures [53], [54]. Since all the computations are run on an individual computer, the large grid simulations for the Sohrab solver are not possible. The rotor–stator interaction simulations have to limited to a 2D mesh.

5.5.1 ERCOFTAC Radial Pump

Recently, an extended version of OpenFOAM (OpenFOAM 1.5–dev) has been developed for turbomachinery applications, and an associated test case of a centrifugal pump including both experimental and numerical data have been published in OpenFOAM wiki website [55]. The viscous and potential flow effects in the small radial gap between rotor and vaned diffusers in the pump have been investigated. The satisfactory agreement between numerical and experimental results

for this test case attracts a number of CFD users to note the further application of OpenFOAM in industrial fields. Hence, taking this test case to validate the Sohrab solver would interest the engineers as well.

5.5.1.1 Geometry of the Pump Stage

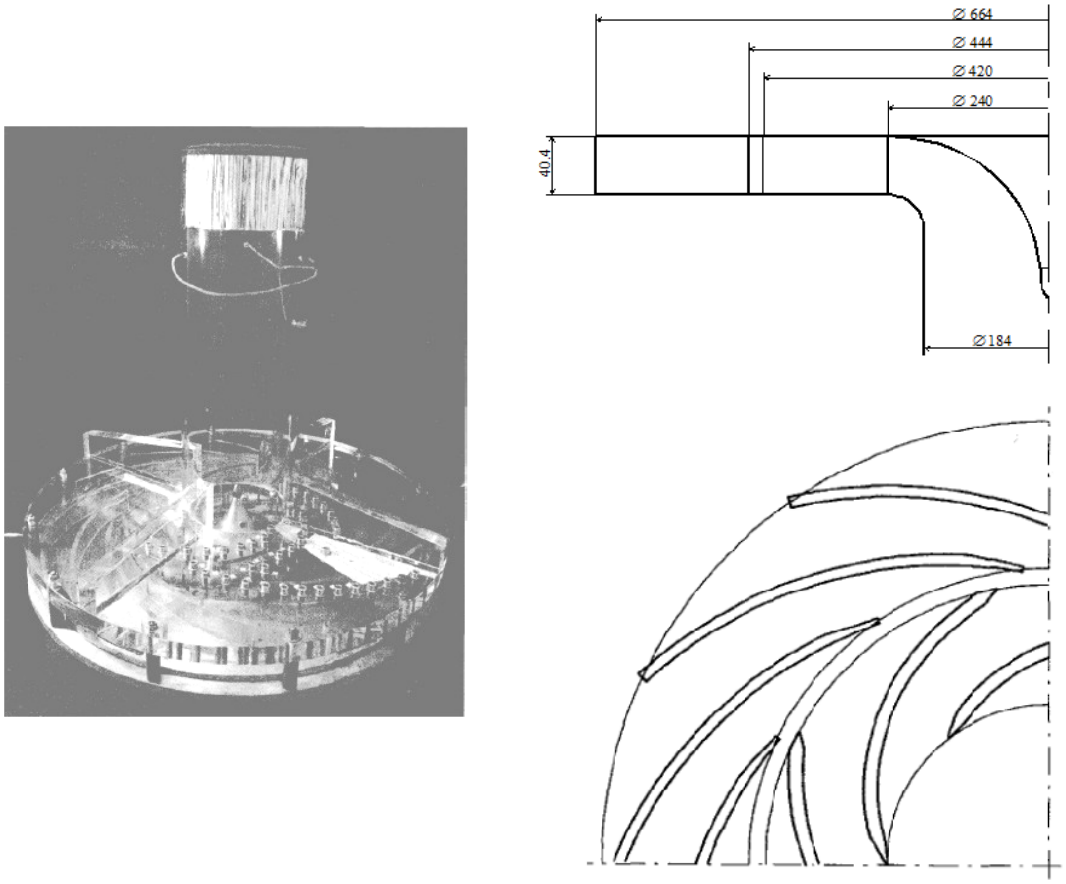


Fig. 5.52 ERCOFTAC pump model [55]

The test model is a centrifugal pump first presented by Combès at a Turbomachinery Flow Prediction ERCOFTAC Workshop [56]. The simplified model of this pump has seven 2D backward impeller blades, 12 2D diffuser vanes and a vaneless radial gap of 6% of the impeller outlet radius. A photo of the pump stage and the geometry of the impeller and vaned diffuser are shown in Fig. 5.52 [55].

As described by Ubaldi et al. [57], [58], the pump operates with air at the nominal operating condition. The specifications of the pump stage are listed in Tab. 5.13.

Impeller			**Diffuser**		
Number of blades	Z_i	7	Number of vanes	Z_d	12
Inlet radius	R_1	120 mm	Inlet radius	R_3	222 mm
Outlet radius	R_2	210 mm	Outlet radius	R_4	332 mm
Blade span	b_i	40.4 mm	Vane span	b_d	40.4 mm
Operating conditions					
Rotating speed	n	2,000 rpm			
Flow rate coefficient	φ	0.048			
Total pressure rise coefficient	ψ	0.65			
Temperature	T	298 K			
Air density	ρ	1.2 kg/m^3			

Tab. 5.13 Specifications of the pump stage [57], [58]

5.5.1.2 Numerical Simulation Setup

Numerical computation of this pump have been done by Bert et al. [59], [60], Torbergsen and White [61], Sato and He [62 – 64], Page and Beaudoin [65], [66], Petit [67] and Xie, [68], including both 2D and 3D simulations. As discussed in [59], relevant information could be recovered by the 2D simulations although the real flow owns 3D features. In the present work only 2D simulation is performed, due of the huge computational cost of the 3D grids under OpenFOAM environment. In order to produce the numerical results which can be directly compared with the preliminary data reported in literatures [57], [68], original grids and boundary conditions described in [68] are applied in the Sohrab solver. The total number of grid nodes for the 2D simulation is about 194,000. The complete structured grid for the computational domain is shown in a midplane in Fig. 5.53, including impeller blades and diffuser vans. In addition, the grids for the impeller and diffuser are shown respectively in Fig. 5.54 and Fig. 5.55 with the details near the leading edge and the trailing edge.

Although the GGI connection between the couple patches increases the computational effort and decreases the accuracy of the result by processing the interpolation between two sides of the interface. It is the only choice to couple the standing and moving (rotating) patches of the mesh in OpenFOAM. The GGI function in OpenFOAM is developed by M. Beaudoin and H. Jasak [69] and only available in the developed versions, e. g. OpenFOAM 1.5–dev and OpenFOAM 1.6–ext. In the present simulation, two submeshes with the impeller and diffuser patches defining the boundary of the moving region are included in the mesh to utilizing the GGI feature.

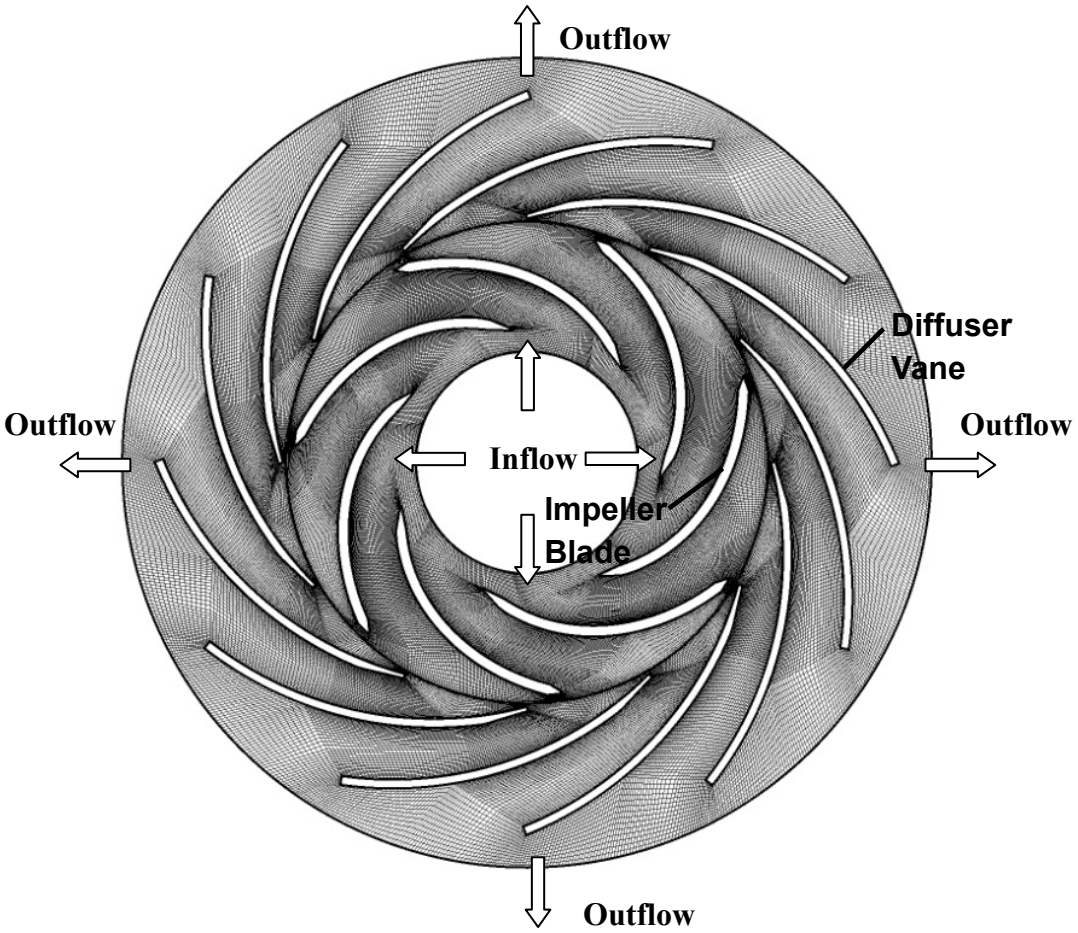


Fig. 5.53 2D mesh of the complete pump stage

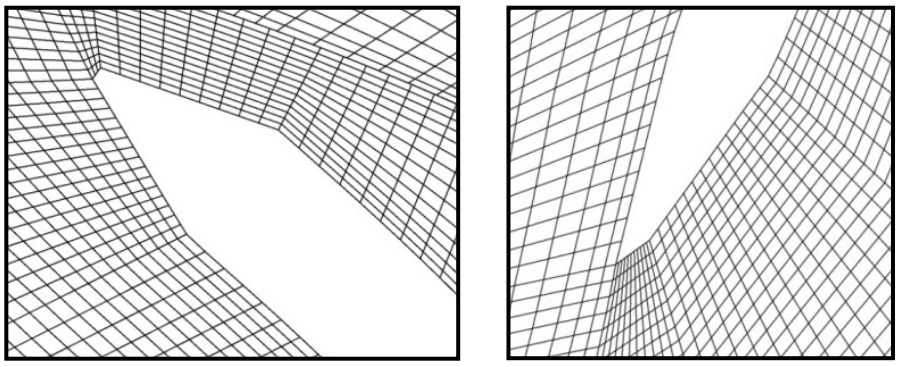


Fig. 5.54 Grids near the both leading and trailing edges of the impeller blade

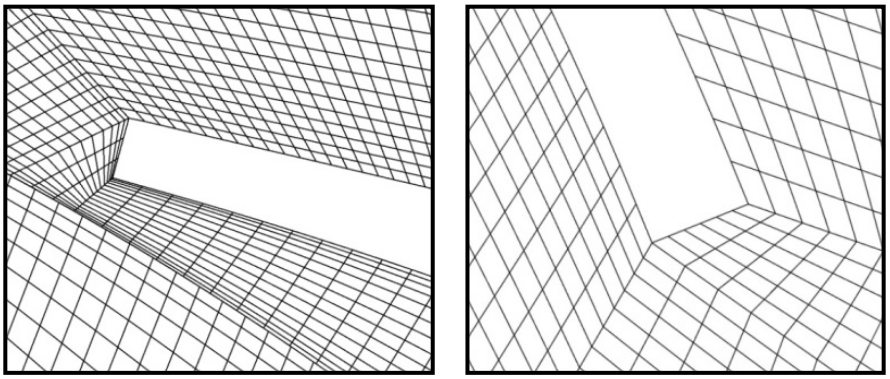


Fig. 5.55 Grids near the both leading and trailing edges of the diffuser vane

Considering the transient flow effect, the dynamic mesh solver is applied to investigate the rotor–stator interaction. The complete pump stage is simulated by the Sohrab solver, including seven impeller blades and 12 diffuser vanes. The library OpenFOAMTurbo available in the developed version of OpenFOAM provides the capability to specify the inflow parameters in a cylindrical coordinate system. The radial velocity is used to define the inlet velocity and the average static pressure is given at the outlet. A no–slip boundary condition is specified at the blades and the vanes. The boundary condition of the hub and shroud is set to empty due to the 2D property of the model. k–ε turbulence model is applied to simulate the turbulence. The average value of y^+ is well controlled to be below 40 for the blades and the vanes. At the inlet the turbulence intensity is set to 5% with the viscosity ratio of 10. The discretisation in space is second order accurate, bounded for the convection component and unbounded for the diffusion component. For the time dependent component, the second order backward implicit scheme is applied. These setups are summarized in Tab. 5.14 and Tab. 5.15.

Inlet	$\mathbf{U}_{in}$ = 10.81 m/s, $\mathbf{w} = \mathbf{U}_{laminar}$, 5% turbulence intensity
Outlet	Average static pressure, p = 0 Pa
Blade & vane	No–slip B. C.
Symmetric walls	Symmetric B. C.
Working fluid	Air, $\nu = 1.0\times10^{-5}$ m^2/s
Turbulence model	k–ε

Tab. 5.14 Boundary conditions setup for the flow in a radial pump

Component:	Scheme:	Numerical behaviour:
Time	Backward	Second order, implicit
Convection	TVD	First/second order, bounded
Diffusion	CD	Second order, unbounded
Pressure gradient	CD	Second order, Gaussian integration

Tab. 5.15 Numerical schemes setup for the flow in a radial pump

Due to the complexity of the mesh interpolation between the static solver result and the dynamic mesh solver result in OpenFOAM, the transient simulation is directly executed for the pump without using any initial flow field. In order to preserve the stability of the numerical computation and trend a better convergence, the courant number is used to control the size of the time step. This parameter is defined as:

$$Co = \frac{\mathbf{U}\Delta t}{\Delta x} \tag{5.15}$$

In the present case, the maximum courant number is limited to 0.5. Then, the associated time step after five impeller revolutions is about 8×10^{-6} s. Comparing to the rotational speed of the impeller n = 2,000 rpm, the rotating angle for each time step is

$$\omega = \frac{2\pi n}{60} \tag{5.16}$$

$$\Delta\varphi = \omega \cdot \Delta t < 0.1^\circ \tag{5.17}$$

Therefore under the courant number control, an extremely fine impeller rotation is obtained at the heavy expense of the computation time, by the small time step. However, this offers a converged solution and satisfactory accuracy for the GGI interpolation in OpenFOAM.

5.5.1.3 Results & Comparisons

In this section, the numerical results obtained by the Sohrab solver are compared with those of the Navier–Stokes solver. Moreover, the velocity components in the radial gap region computed from the both solvers are validated by the measured data.

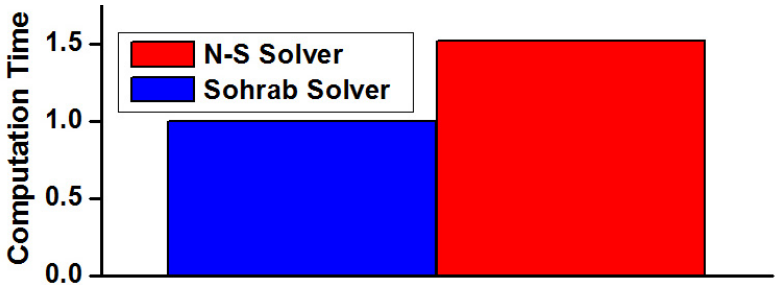
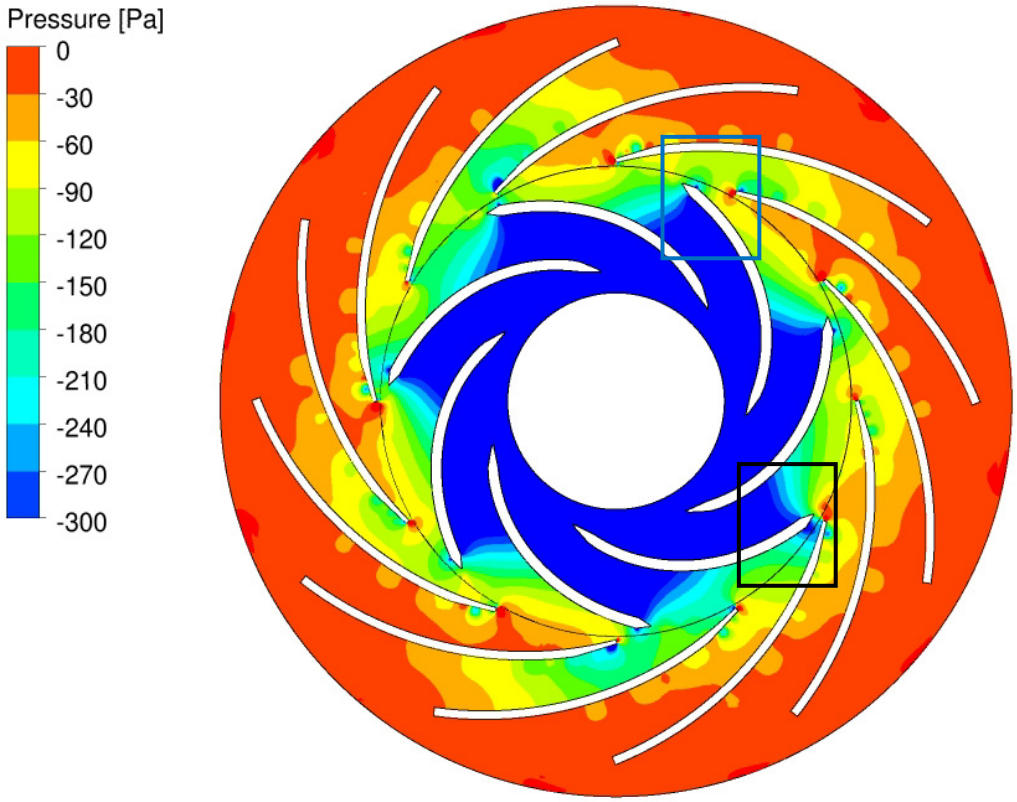


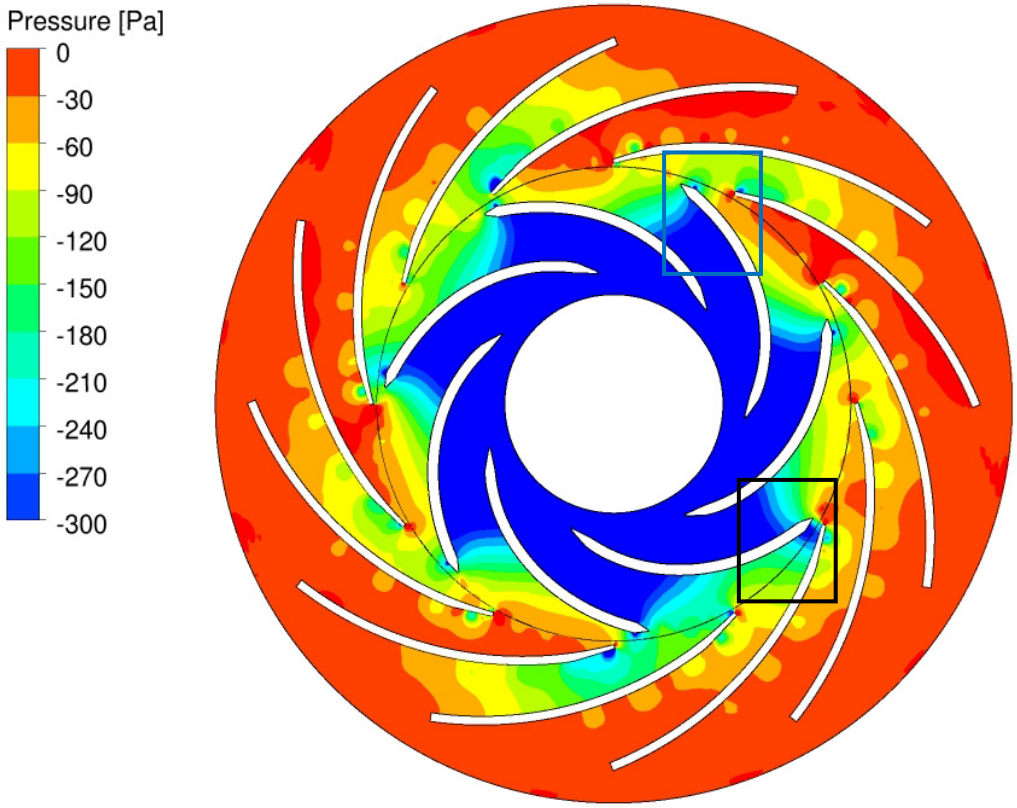
Fig. 5.56 Comparison of computation time taken by both solvers

After five impeller revolutions, the two solvers solutions converge. However, in order to reach periodic flow in the results, another five impeller revolutions are carried out. A comparison of the total computation time for 10 impeller revolutions taken by the two solvers is shown in Fig. 5.56. The Navier–Stokes solver is only about 1.5 times slower than the Sohrab solver in completing the simulation. This result coincides with the analysis discussed in section 4.3 that the dynamic mesh solver for the modified equation of fluid motion requires more computational time, as compared to the static mesh solver, however, the Sohrab solver would be still faster than the Navier–Stokes solver.

The results obtained from the last impeller revolution are compared. The static pressure distributions in the whole pump stage computed by the both solvers are presented in Fig. 5.57. It can be confirmed that, the both solvers show very similar results. In the radial direction, a strong pressure gradient can be observed. The pressure distribution is nearly periodic for each blade in the impeller. However downstream the blades, the rotor–stator interaction strongly affects the pressure distribution in the diffuser, thus resulting in unsymmetrical pressure distribution due to the unsymmetrical geometry of the pump stage. In order to investigate the pressure distribution in the blade tip regions which are marked in Fig. 5.57, these regions are enlarged in Fig. 5.58. As shown in the figure, because of the strong rotor–stator interaction, the minimum pressure is no longer at the blade suction side surface and the shed wake from the blade is directed to the pressure side of the neighboring blade. This phenomenon is more obvious when the blade is close to the vane.



Navier–Stokes solver



Sohrab solver

Fig. 5.57 Comparison of the static pressure distribution

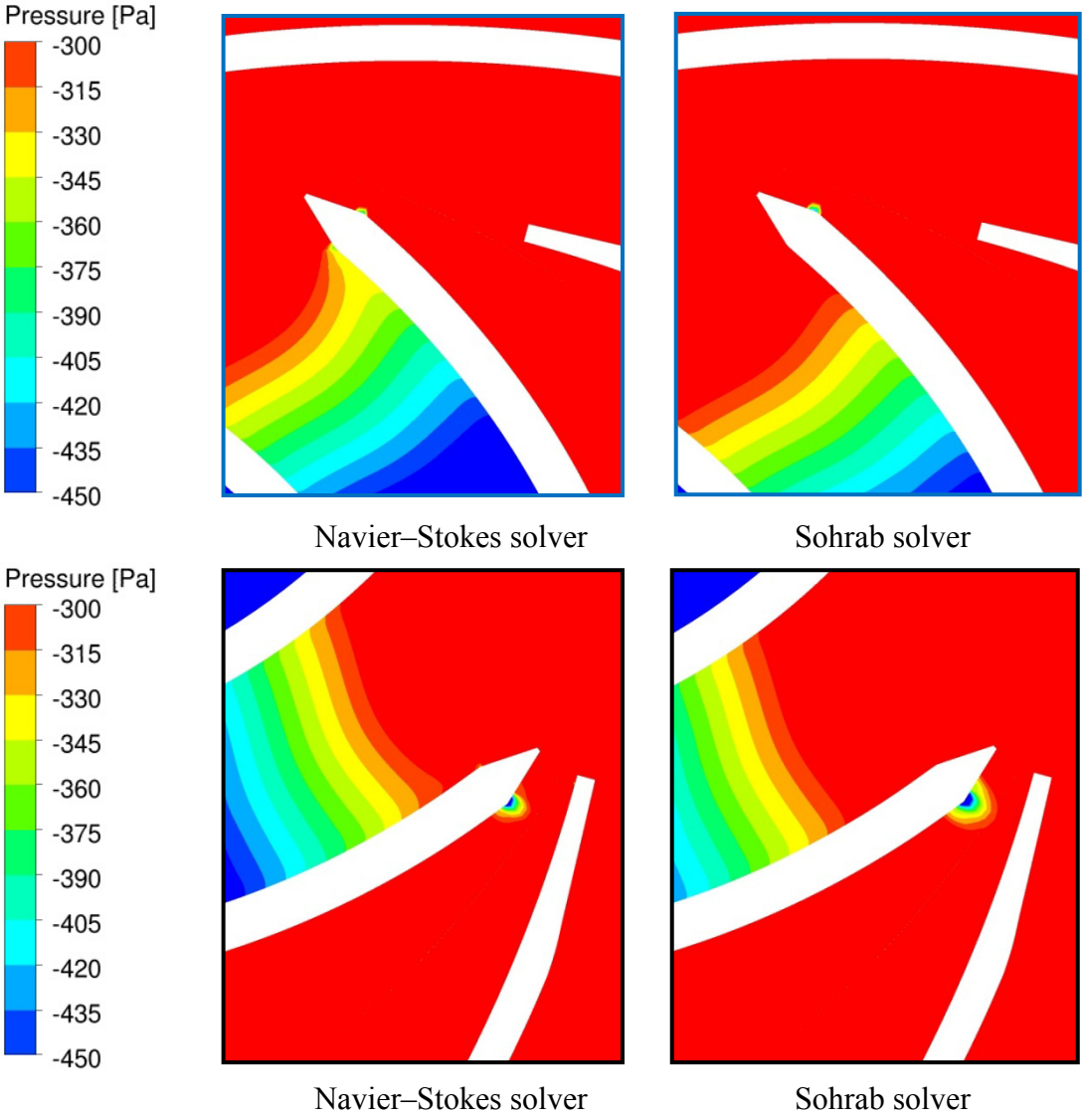


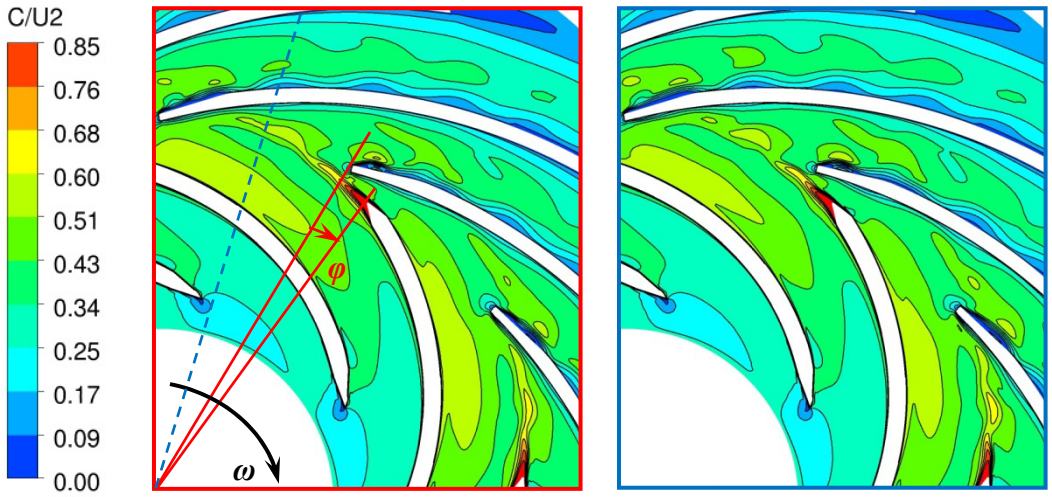
Fig. 5.58 Comparison of the static pressure distribution in the enlarged blade tip region

The flow field is also compared including the absolute velocity **C** obtained by the numerical solvers and the relative velocity **W** obtained by both numerical and experimental methods. The relative velocity **W** is based on the reference of the rotating impeller by subtracting the local circumferential velocity **U** from the absolute velocity **C** as calculated in Eq. (5.18):

$$\mathbf{W} = \mathbf{C} - \mathbf{U} \tag{5.18}$$

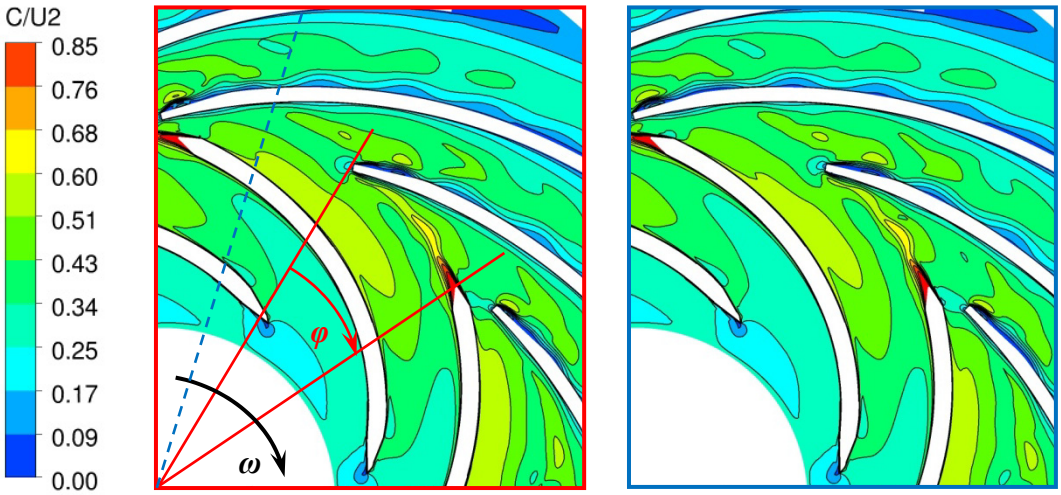
Corresponding to the measurement position, the absolute velocity fields at four different impeller positions are shown qualitatively in Figs. 5.59 – 5.62, for the results obtained from both Sohrab and Navier–Stokes solvers. As marked in the figure, the impeller rotates in clock–wise direction and the positions are defined by the angle

between the blade trailing edge and vane leading edge. The comparisons present a satisfactory agreement for the whole flow region, although the Sohrab solver predicts slightly smaller velocity at the blade pressure side as compared with the Navier–Stokes solver. Both numerical results present that the diffuser vanes have only limited influence on the impeller inner velocity field while strongly disturb the rear part of the impeller flow and the wake shed by the blade at all shown impeller positions. In the rear part region, the velocity field is greatly influenced by the relative impeller position to the diffuser vane leading edge. In the wake region near the trailing edge of the blade, a high absolute velocity is developed because for a backward blade the relative velocity is low at the outlet of the impeller. This result can be illustrated by the velocity triangles at the impeller outlet, as sketched in Fig. 5.63. All these flow phenomena coincide with the physical observation of the rotor–stator interaction.



Navier–Stokes Solver Sohrab Solver

Fig. 5.59 Comparison of the absolute velocity fields at $\varphi = 5°$



Navier–Stokes Solver Sohrab Solver

Fig. 5.60 Comparison of the absolute velocity fields at $\varphi = 25°$

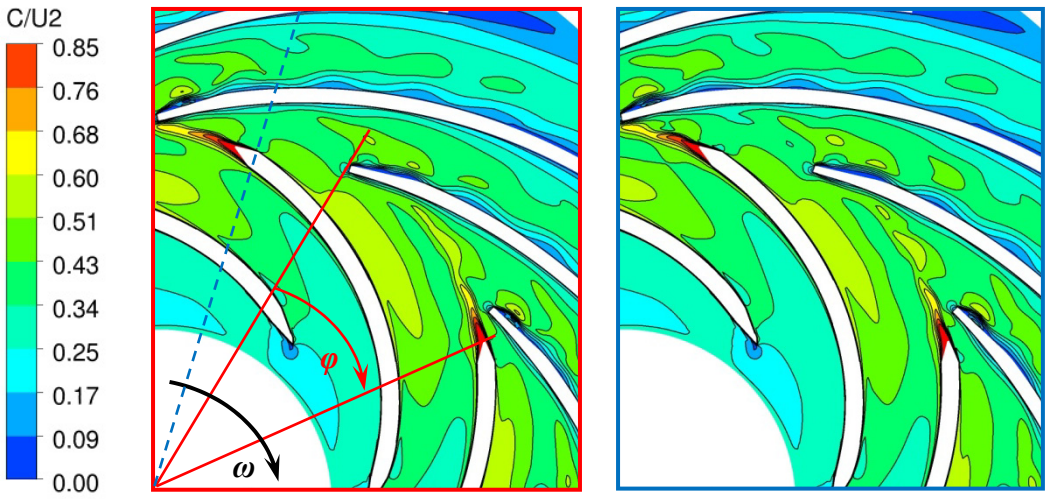


Fig. 5.61 Comparison of the absolute velocity fields at $\varphi = 35°$

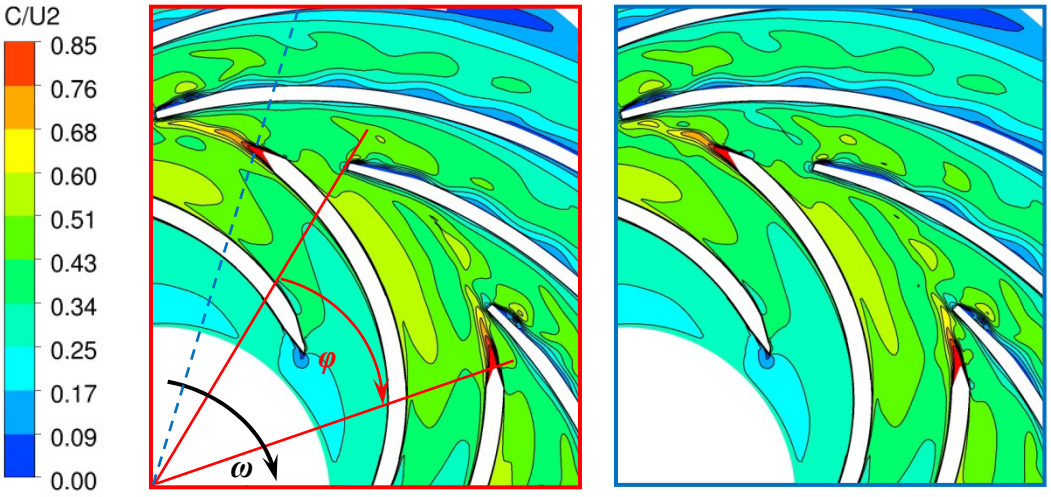


Fig. 5.62 Comparison of the absolute velocity fields at $\varphi = 40°$

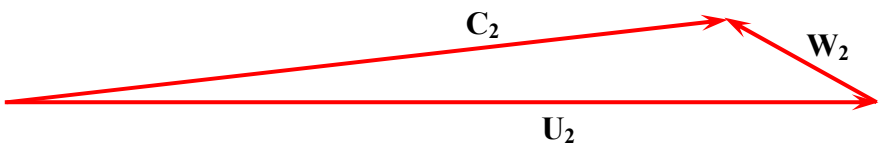


Fig. 5.63 Velocity triangle at the impeller outlet

The strongest rotor–stator interaction occurs in the radial gap region between the impeller trailing edge and the diffuser leading edge. The computed relative velocity components in radial and circumferential direction in this most interesting region are

compared with the measured data reported in [55]. The two relative velocity components can be calculated from the absolute velocity component.

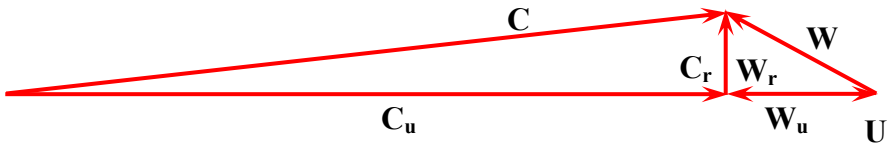


Fig. 5.64 Circumferential and radial components of the velocities

Considering the velocity triangle sketched in Fig. 5.64, the following equations are obtained:

$$\mathbf{W}_u = \mathbf{C}_u - \mathbf{U} \tag{5.19}$$

$$\mathbf{W}_r = \mathbf{C}_r \tag{5.20}$$

Figs. 5.65 – 5.68 present the comparisons of the relative velocity components at r/R_2 = 1.02 in the radial gap region, for four aforementioned impeller positions between the measurement and numerical computations. For clarity, the circumferential positions of the impeller blades and diffuser vanes are respectively marked at the top and bottom of the figures in order to illustrate a two pitches circumferential distance for the horizontal axis. As compared with Figs. 5.59 – 5.62, Figs. 5.65 – 5.68 show the same impeller positions however only partly cover the flow regions. In detail, the relative velocities are sampled from the gap near the four diffuser vanes below the blue line sketched in Figs. 5.58 – 5.61, due to limit of the measured data.

The comparisons indicate a satisfactory agreement between the measurement and numerical simulations, especially in the circumferential direction. Both numerical results match each other very well except few peak and valley regions. Referring to Eq. (5.19) and Eq. (5.20), the slight difference between the results of the Sohrab and Navier–Stokes solvers coincide with the absolute velocity comparisons in Figs. 5.59 – 5.62. The quantitative comparisons of the relative velocity components present a better understanding of the effect induced by the rotor–stator interaction. Series peaks and valleys can be observed in the relative velocity distributions, occurring in the vicinity of the impeller blade trailing edges and diffuser vane leading edges. The velocity gradient is relatively small in the portion where no blade or vane is presented. These phenomena are obvious at impeller positions φ = 25° and φ = 35°. Near the impeller blade the velocity decreases in the wake region of the suction side while increasing at the pressure side. Further, the diffuser vanes also make strong influence

to the impeller outlet flow, thus producing the velocity valley near its leading edge. Especially when the flow downstream the blade directly encounters a vane, the velocity experiences a very high gradient. As a consequence, the velocity components in the gap region strongly depend on the impeller relative position to the diffuser [57].

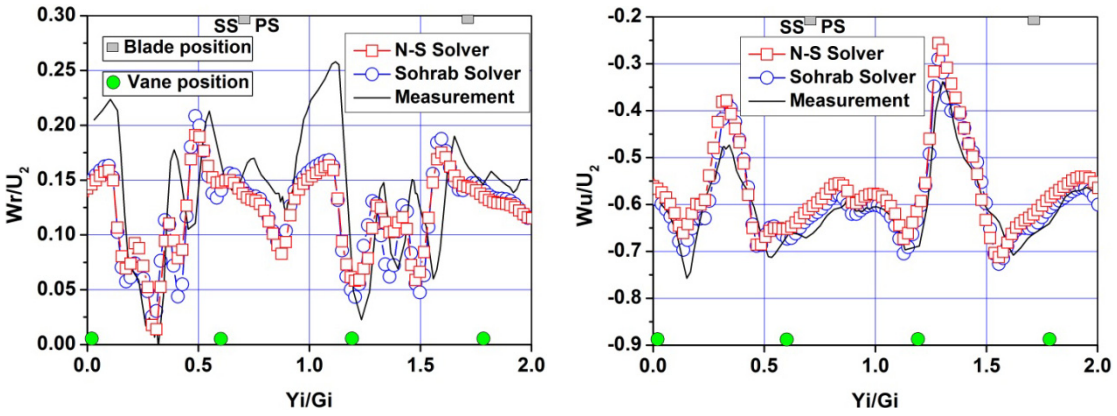


Fig. 5.65 Comparison of relative velocity components in radial gap region at $\varphi = 5°$

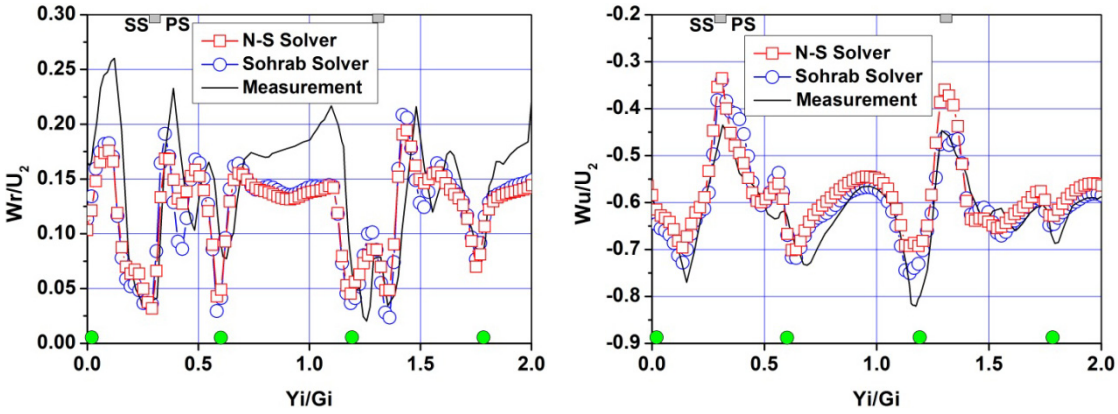


Fig. 5.66 Comparison of relative velocity components in radial gap region at $\varphi = 25°$

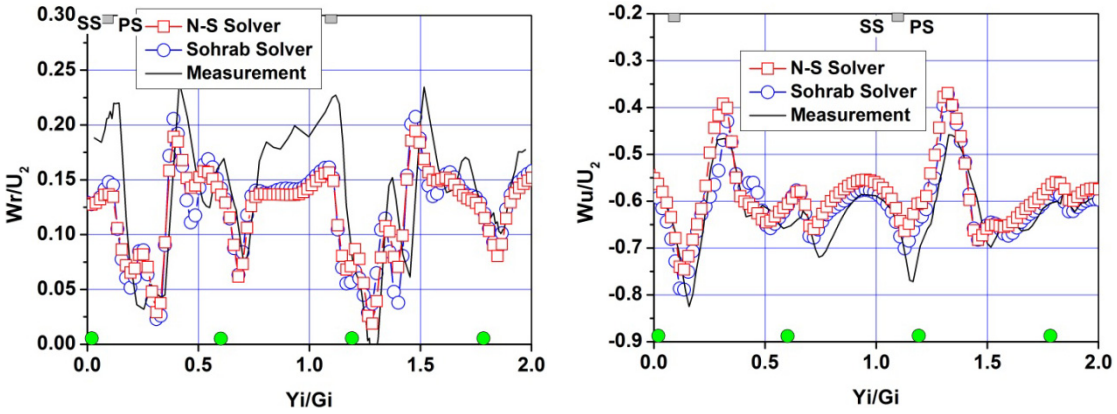


Fig. 5.67 Comparison of relative velocity components in radial gap region at $\varphi = 35°$

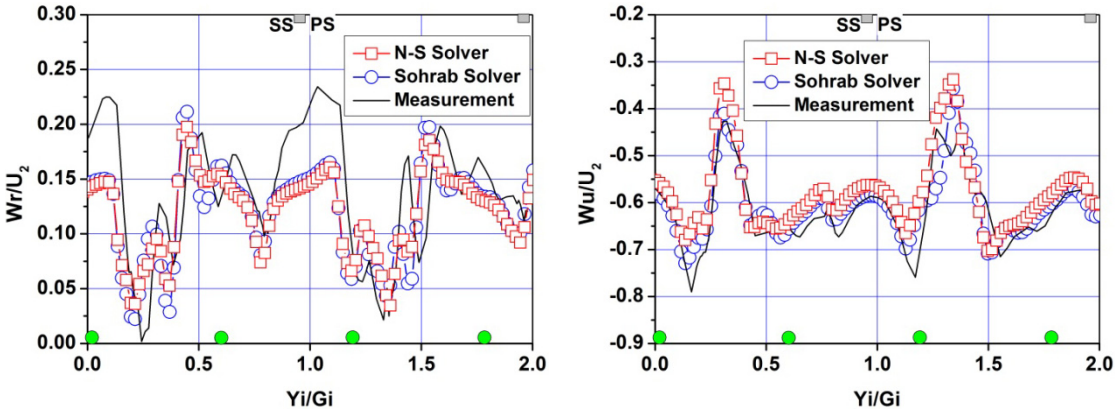


Fig. 5.68 Comparison of relative velocity components in radial gap region at $\varphi = 40°$

5.5.2 Symmetrical Radial Pump

The dynamic mesh function of the Sohrab solver has been examined by an open case in the section above. In order to further assess the ability of the Sohrab solver in simulating complex flow conditions, another symmetrical radial pump, which has been investigated by Feng [54], is taken for the simulation. The obtained results of the solver are evaluated by comparing them with those obtained by other methods, including the numerical results obtained by the Navier–Stokes solver and the measured data.

5.5.2.1 Geometry of the Pump Stage

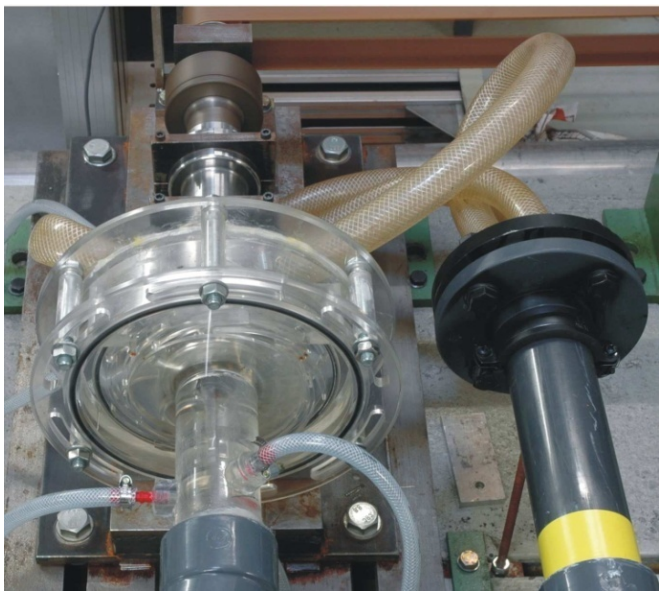


Fig. 5.69 Pump view [54]

The pump stage of the radial pump is shown in Fig. 5.69. As described by Feng [54], the impeller is shrouded with six 2D blades, the radial gap between the impeller outlet and diffuser inlet is 2.25 mm corresponding to 3% of the impeller outlet radius and the diffuser owns nine 2D vanes. The specifications of the pump stage are listed in Tab. 5.16.

Impeller			**Diffuser**		
Number of blades	Z_i	6	Number of vanes	Z_d	9
Inlet radius	R_1	40 mm	Inlet radius	R_3	77.5 mm
Outlet radius	R_2	75.25 mm	Outlet radius	R_4	95 mm
Blade span	b_i	12.7 mm	Vane span	b_d	14 mm
Design Operating conditions					
Rotating speed	n	1,450 rpm			
Flow rate	Q	0.0045 m^3/s			
Delivery head	H	7 m			

Tab. 5.16 Specifications of the pump stage [54]

5.5.2.2 Numerical Simulation Setup

Due to the 2D geometry property of the blades and the huge computational cost for the 3D simulation, the pump stage is simplified as a 2D model and only the impeller and diffuser are taken into account for the numerical simulation as illustrated in Fig. 5.70. The 2D structured grid for the flow model is created by utilizing the grid generation tool ICEM–CFD. The resultant mesh is converted to the fluent format and loaded into OpenFOAM by the MeshToFoam interface tool. The total grid number for the computational domain is about 150,000, including six impeller blades and nine diffuser vanes. GGI is used to couple the rotating blades and standing vanes. For the surrounding of the blades and vanes, an O–Grid is built up to give a satisfactory resolution for the boundary layer near them. While for the blade passage, an H–grid is applied. The detailed grid information near the leading and trailing edges of the blades and vanes are enlarged in Fig. 5.71 and Fig. 5.72 respectively for better views.

The dynamic mesh solvers including both Sohrab and Navier–Stokes, are applied to simulate the pump stage. The same grid and boundary conditions are applied to the both solvers to give an equivalent evaluation for the results. The boundary conditions for the transient simulations are listed in Tab. 5.17.

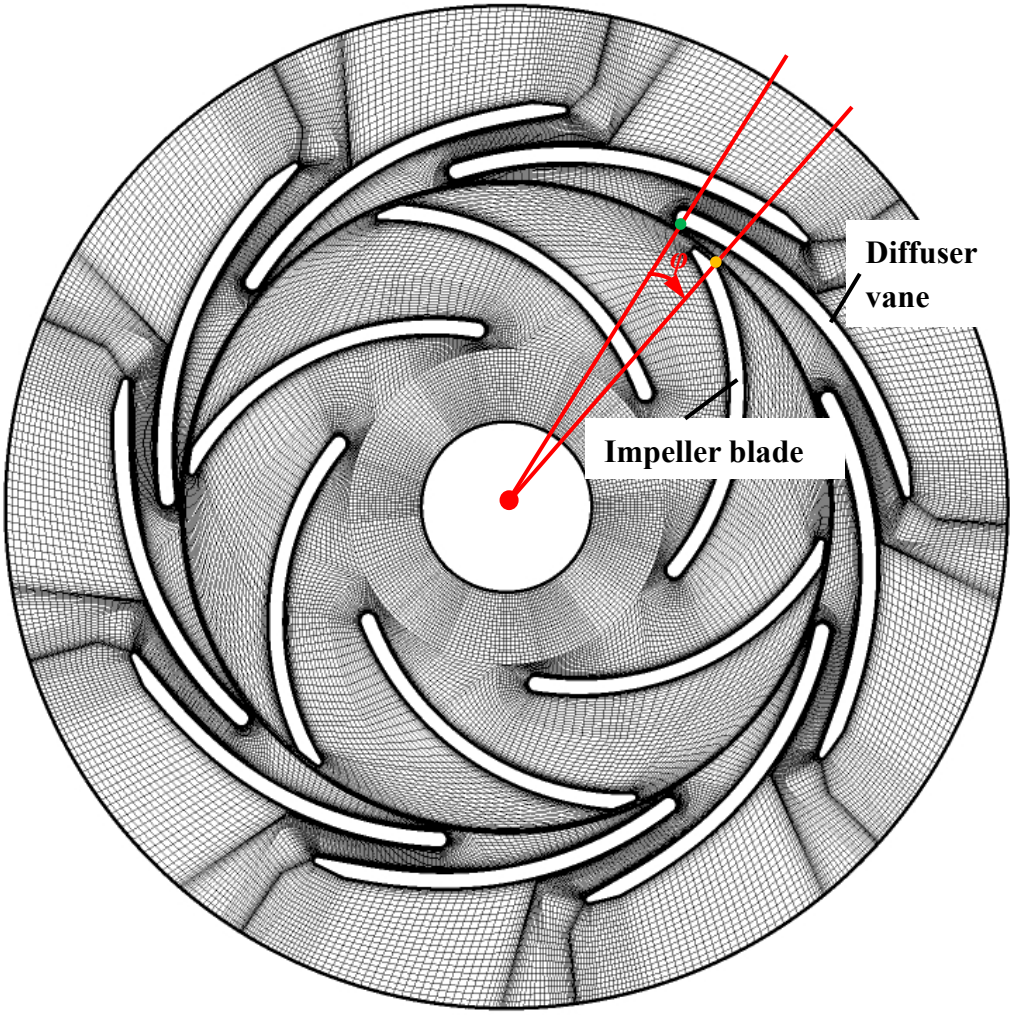


Fig. 5.70 2D mesh of the complete pump stage

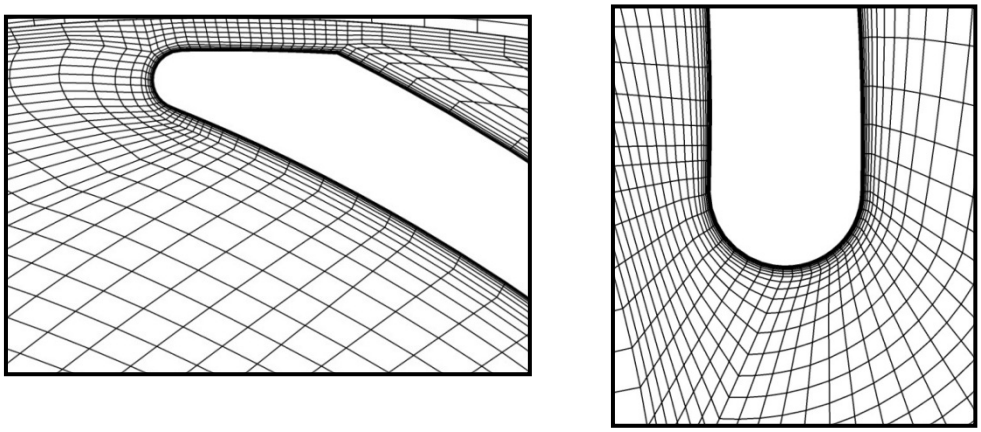


Fig. 5.71 Grids near the both leading and trailing edges of the impeller blade

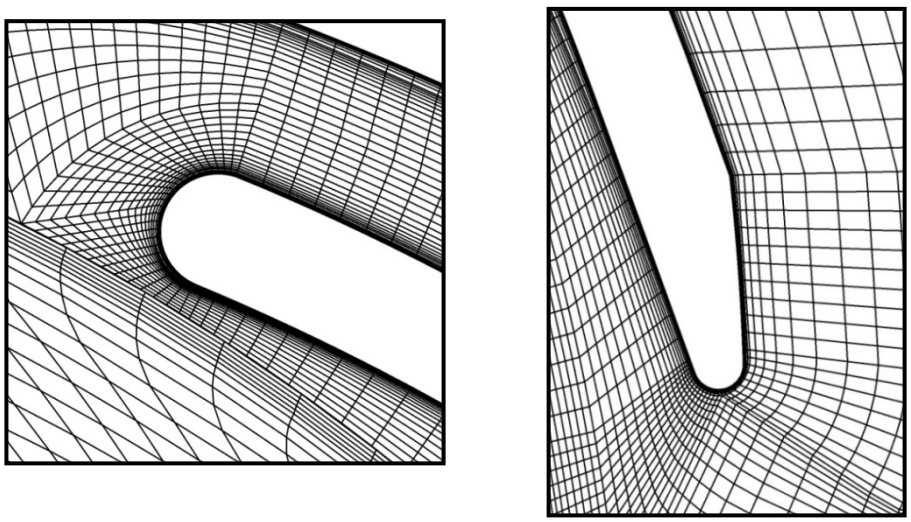


Fig. 5.72 Grids near the both leading and trailing edges of the diffuser vane

Inlet	$\mathbf{U}_{in}$ = 2.819 m/s, $\mathbf{w} = \mathbf{U}_{laminar}$, 5% turbulence intensity
Outlet	Average static pressure, p = 0 Pa
Blade & vane	No–slip B. C.
Symmetric walls	Symmetric B. C.
Working fluid	Water, $\nu = 8.94\times10^{-7}$ m^2/s
Turbulence model	*SST*

Tab. 5.17 Boundary conditions setup for the flow in a symmetric radial pump

The numerical schemes for the present pump simulation are the same as those listed in Tab. 5.15. Similar to the time step setup for the ERCOFTAC radial pump simulation, the courant number is set below 0.5, $Co < 0.5$, to preserve the stability of the numerical computation in OpenFOAM, thus resulting in a rotating angle less than 0.1 deg, $\Delta\varphi < 0.1^{o}$, for each time step. Comparing to the CFX setup in [54], which uses $\Delta\varphi = 1^{o}$, it is found that OpenFOAM needs a much finer rotating angle for the simulation, thus resulting in a large computation time consuming.

5.5.2.3 Results & Comparisons

In Fig. 5.73, the computation time taken by both Sohrab and Navier–Stokes solvers are compared. As expected, the Sohrab solver saves about one third computation time as compared to the Navier–Stokes solver, which is similar to the result in Fig. 5.56.

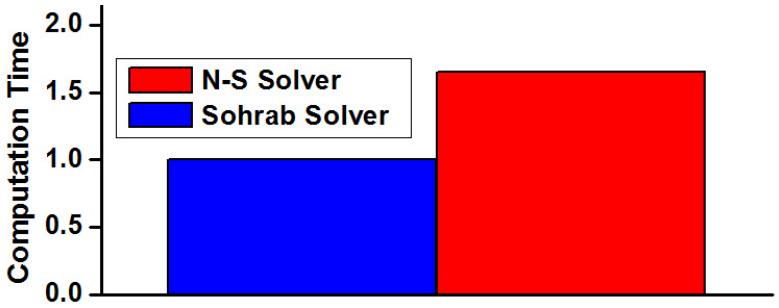
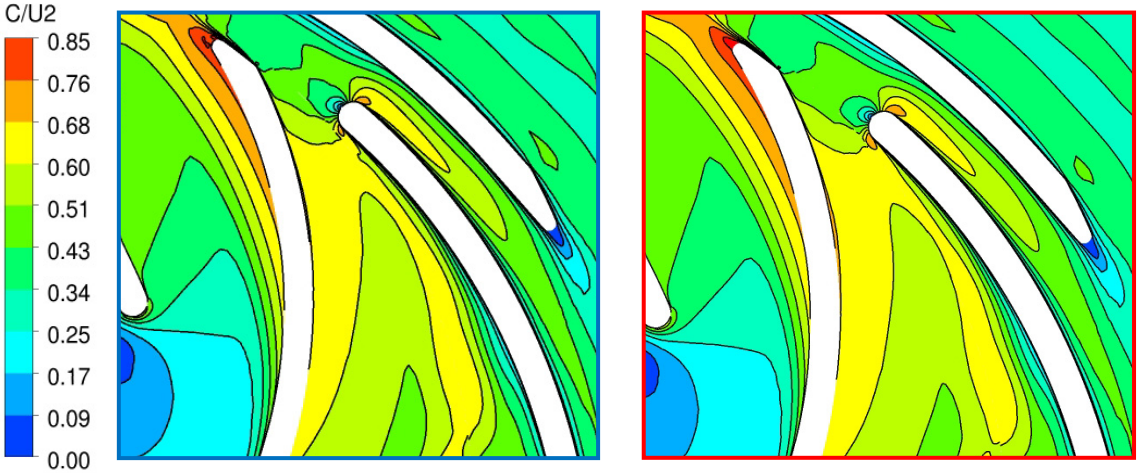


Fig. 5.73 Comparison of computation time taken by both solvers

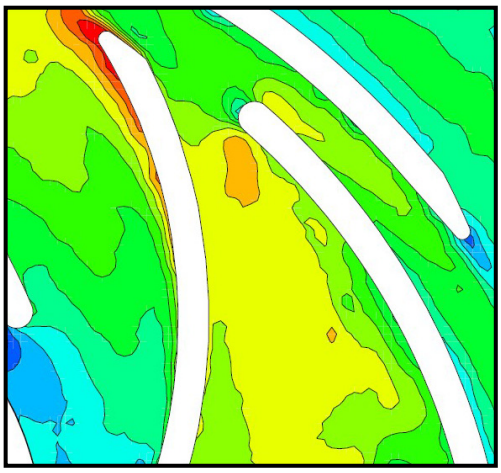
The flow fields of the absolute velocity **C** at three impeller positions, $\varphi = -10°$, $\varphi = 0°$ and $\varphi = 10°$ are presented qualitatively in Figs. 5.74 – 5.76. The results obtained by both Sohrab and Navier–Stokes solvers and experimental results reported in [54] are compared. A satisfactory agreement between the results obtained by both solvers can be asserted. As compared with the experimental results, the numerical results represent the flow development affected by the impeller position, not only at the suction side but also at the pressure side of the diffuser vane. Near the leading edges of the diffuser vanes, a high velocity region at the suction side, is strongly influenced by the rotating blade of the impeller, thus decreasing from $\varphi = -10°$ to $\varphi = 10°$, meanwhile, at the pressure side, a smaller high velocity region also decreases gradually.

Then, the relative velocities in the radial gap region in these figures are quantitatively compared in Figs. 5.77 – 5.79. It can be observed that the velocities calculated by both solvers coincide well with each other. Comparing them with the measured data reported by Feng [54], generally speaking, it can be found that numerical results are in good agreement with the measured data, although the numerical solutions predict slightly smaller values than the measurement which is more obviously in the circumferential direction. The results comparisons in this section show the similar situations as discussed in the section 5.5.1. Such results offer further evidence to prove the accuracy of the Sohrab solver.



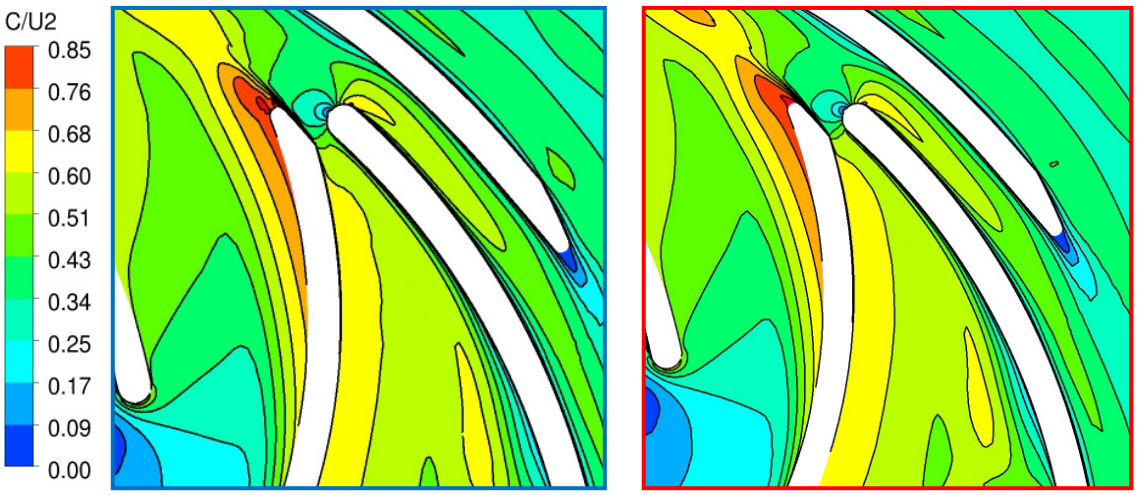
Sohrab Solver

Navier–Stokes Solver

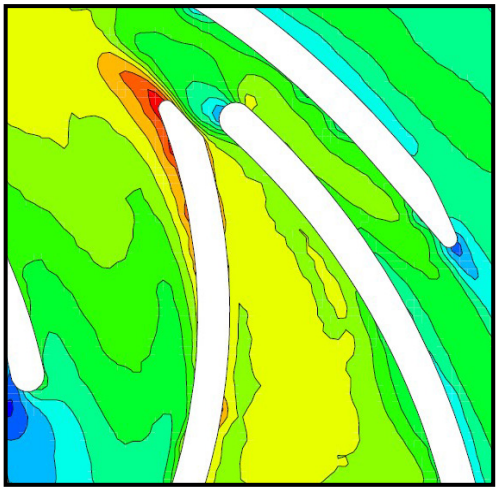


PIV Measurement

Fig. 5.74 Comparison of the absolute velocity fields at φ = -10°

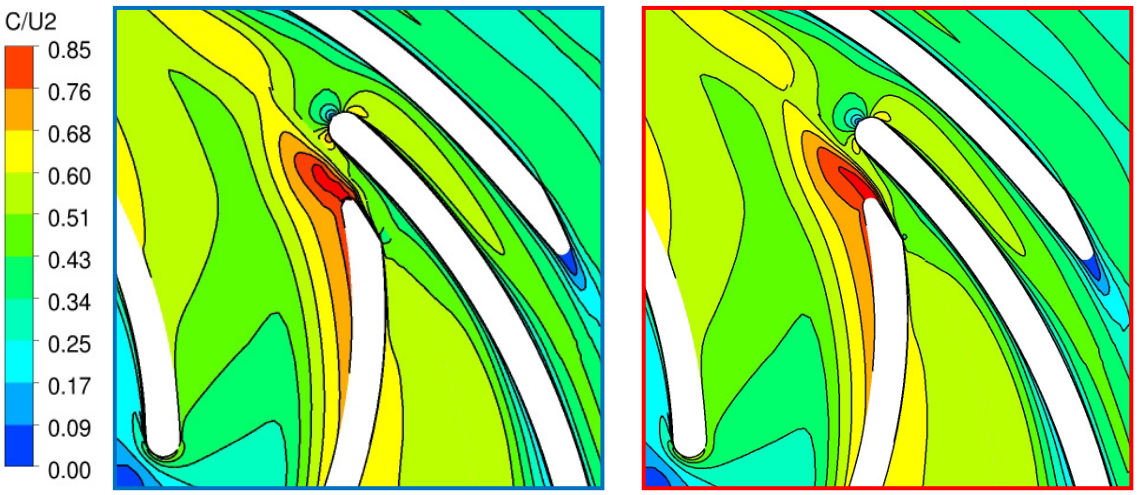


Sohrab Solver Navier–Stokes Solver



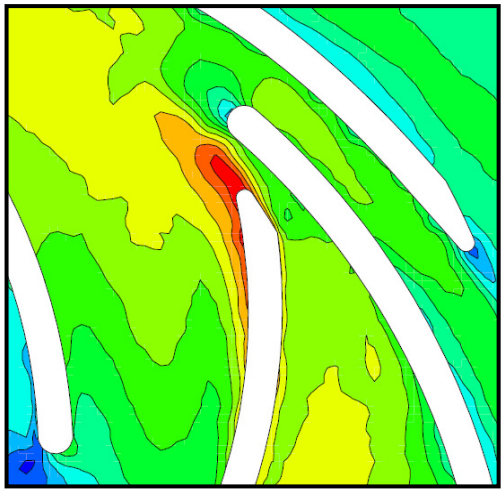
PIV Measurement

Fig. 5.75 Comparison of the absolute velocity fields at $\varphi = 0°$



Sohrab Solver

Navier–Stokes Solver



PIV Measurement

Fig. 5.76 Comparison of the absolute velocity fields at $\varphi = 10°$

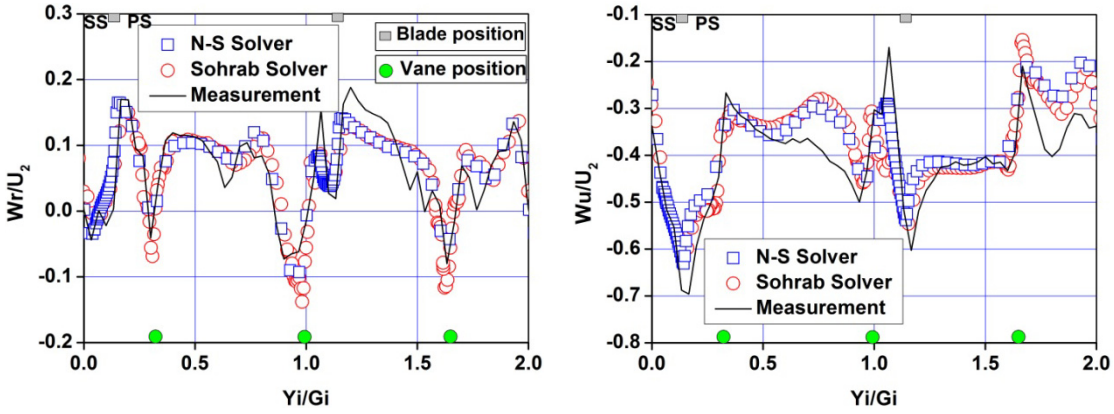


Fig. 5.77 Comparison of relative velocity components in radial gap region at φ = -10°

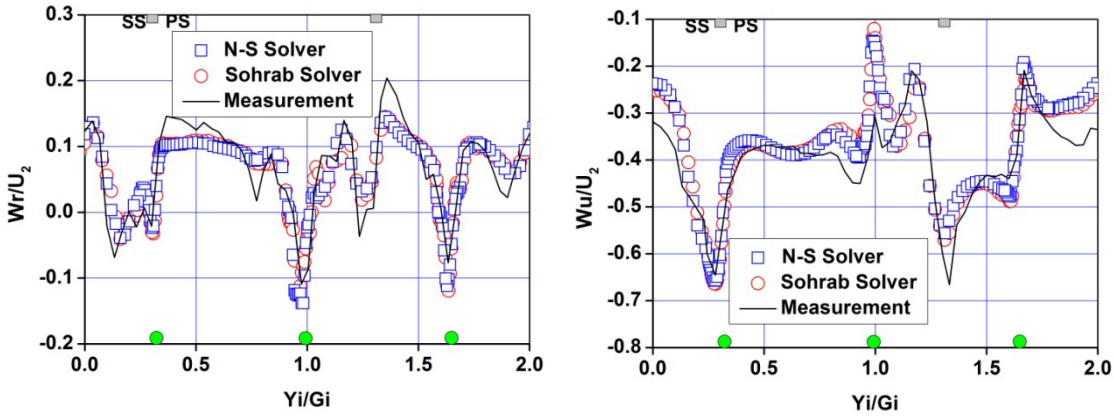


Fig. 5.78 Comparison of relative velocity components in radial gap region at φ = 0°

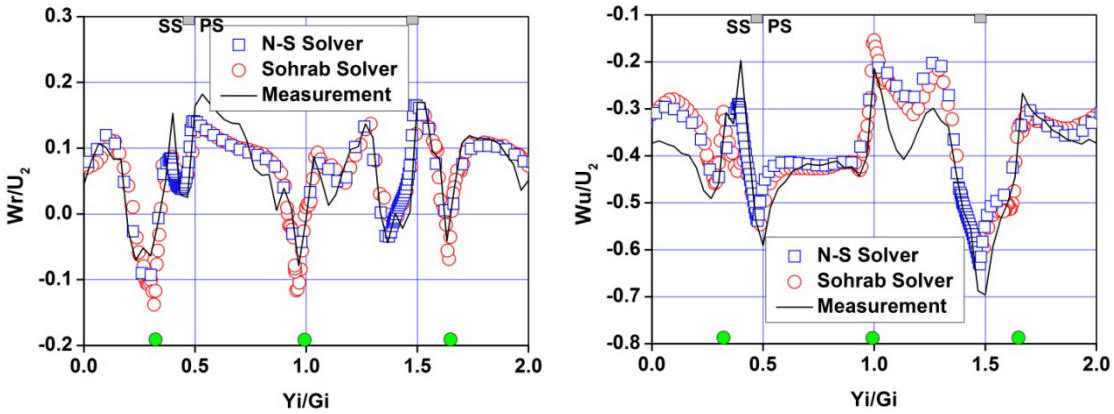


Fig. 5.79 Comparison of relative velocity components in radial gap region at φ = 10°

6 Conclusions

This research is mainly focused on the modified equation of fluid motion for incompressible flow and its application in turbomachinery flow. In mathematics, the appearance of the convective velocity in the equation eliminates its non–linearity, as compared to the Navier–Stokes equation. In order to declare the linear advantage of the modified equation in CFD applications, the derivation process, the FVM discretisation results and the solver schemes of the equation have been discussed in detail by comparing them to those of the Navier–Stokes equation. Then, a novel Sohrab solver developed on the basis of the modified equation of fluid motion for incompressible flow is examined for five flow models, including the models for fundamental boundary layer flow and the advanced transient flow in radial pumps. Several important numerical issues of interest have arisen. The results are evaluated by comparing them with those obtained by other methods, including the analytical solution of the modified equation of fluid motion, the numerical results obtained by the Navier–Stokes solver and measured data reported in literatures. In summary, some primary contributions of the research to the developed solver can be distinguished: 1) derivation of the convective velocity, 2) accuracy, 3) numerical stability, 4) numerical errors, 5) mesh resolution requirement, and 6) computation time.

1) The convective velocity plays a significant role in the Sohrab solver because it not only eliminates the non–linearity of the resultant algebraic equations after the FVM discretisation is carried out, but also introduces a fixed parameter to the solver instead of another flow variable. It can be considered as the essential difference between the Sohrab solver and the Navier–Stokes solver. In this context, the derivation process of the convective velocity has been discussed in detail. According to the scale–invariant theory of statistics, at potential flow scale, the vanishing of the viscosity of a fluid makes the modified equation of fluid motion transfer to the Euler equation and the convective velocity become the local velocity. Therefore, the determination of the convective velocity is sourced from the potential scale by calculating the local velocity using Euler equation, further developed in the laminar flow scale, for the solver application in the turbulent flow scale. All the Sohrab solver results obtained for the turbulent flow simulations in this research support this numerical method to determine the convective velocity.

2) The primary function of a numerical solver is to deliver accurate numerical solutions by solving the equation using iteration methods. In terms of the analytical solution of the modified equation of fluid motion, the Sohrab solver is confirmed to be able to provide the numerical solution precisely for the modified equation of fluid motion. Further, the comparisons with the experiments prove that

the Sohrab solver produces physically reasonable results validated by measured data for turbomachinery applications.

3) After the FVM discretisation is carried out, a fixed convective velocity in the modified equation of fluid motion provides high stability to the numerical solution for the Sohrab solver thus avoiding the numerical oscillations of not only the velocity but also the pressure. This high stability contributes to reduce the mesh resolution requirement and the computation time of the solver.

4) Most FVM schemes introduced during the equation discretisation process result in discretisation errors in numerical solutions, influencing the order of accuracy of the solvers. By focusing on the numerical diffusion derived from the convection component, this research has investigated the numerical errors using the analytical and numerical methods. Both methods prove that in terms of the linear advantage, the Sohrab solver preserves numerical errors more stable in its numerical solution, thus producing results with higher accuracy, as compared to the Navier–Stokes solver.

5) An adequate mesh is the most important part of a numerical simulation, as the convergence of the calculation and the accuracy of the solution are strongly dependent on its quality. The mesh resolution requirement for the Sohrab solver has been discussed for both 2D and 3D flow problems. Compared to the Navier–Stokes solver, it is found that the high stability, stable numerical error and accuracy of the Sohrab solver allow it to be used with a lower quality mesh.

6) The Sohrab solver, as compared to the Navier–Stokes solver, can considerably reduce the computation time consuming by its simplified algorithm and stable numerical error when carrying out numerical simulations. For most investigated cases in Chapter 5 the computational effort of the Sohrab and Navier–Stokes solvers is emphasized. Each of them confirms the efficiency of the Sohrab solver that it requires less than half the computation time of the Navier–Stokes solver. Except the dynamic mesh solver applications, comparing the computation time for the 3D cascade (Fig. 5.45) with those of the other cases, it is found that for more complicated boundary conditions, the Sohrab solver is more efficient. It is plausible because the complicated boundary conditions in the flow enlarge the numerical error in the Navier–Stokes solver, thus resulting in more computation time consuming for the convergence of the numerical solution. This result demonstrates the scientific credibility of the Sohrab solver for application to turbomachinery flows. Further, if it is extended to design processes in the turbomachinery industry, significant improvements in the design cycle cost and time can be reached.

From a computational point of view, this research also can be considered as a valuable document for the application of the open source coding toolbox OpenFOAM. Most functions of OpenFOAM related to turbomachinery flow field simulations are examined and developed during the research. The results offer another evidence for the accuracy and efficiency of OpenFOAM in turbomachinery flow applications.

On the other hand, the application of the modified equation of fluid motion in this research implicates a connection between the macro and micro world. And this may lead to a better understanding by human beings of the observed physical phenomena in the living world, clearly benefiting the community.

7 Outlook

This research is intended to provide an in–depth study of the modified equation of fluid motion for incompressible flow and its application in turbomachinery applications by using the numerical methods. However the numerical coding tool for the modified equation is so limited that it is not possible to include more interesting 3D turbomachinery flow models. In the future, if new official versions or developed versions of OpenFOAM could manage more computer resources, more 3D flow simulations using the Sohrab solver would be considered to increase the accuracy of the numerical solutions further.

On the other hand, this research is limited in the incompressible flow field. The modified equation of motion and the scale–invariant form of the conservation equations are developed for all flow fields. The present results will stimulate the application of these equations in compressible flow field by using the numerical methods. However, a better understanding of the scientific meaning of the conservation equations has to be delivered and more analytical results are still required for evaluations.

Bibliography

[1] Sohrab, S. H., 2008, "Derivation of Invariant Forms of Conservation Equation from the Invariant Boltzmann Equation," *Proc. 5th WSEAS Int. Conf. on Fluid Mechanics*, A. C. Benim et al., eds., WSEAS Press, Athens, pp. 27–35.

[2] Sohrab, S. H., *The Physical Foundation of the Canonical Grand Unified Theory of Fields and Its Application to Cosmology, Hydrodynamics, Electrodynamics and Chromodynamics*, 1992.

[3] Sohrab, S. H., 1999, "A Scale–Invariant Model of Statistical Mechanics and Modified Forms of the First and the Second Laws of Thermodynamics", Int. J. Therm. Sci., **38**, pp. 845–853.

[4] Sohrab, S. H., 1999, "Modified Form of the Equation of Motion and Its Solution for Laminar Flow over a Flat Plate and through Circular Pipes and Modified Helmholtz Vorticity Equation", *Chemical and Physical Processes in Combustion*, Fall Technical Meeting, The Eastern States Section, pp. 375–378.

[5] Blasius, H., 1908, "Grenzschichten in Flüssigkeiten mit kleiner Reibung", *Z. Math. Phys.* **56**, pp. 1–37.

[6] Schlichting, H., *Boundary Layer Theory*, McGraw Hill, New York, 1968.

[7] Benra, F. K., and Sohrab, S. H., 2006, "Modified Theories of Laminar Flow around a Rigid Cylinder and Flow outside and inside a Liquid Cylinder in Uniform and Counterflow Gaseous Jets", *WSEAS Trans. Fluid Mech.,* Issue 6, **1**, 533.

[8] Sohrab, S. H., 2008, "A Modified Scale Invariant Statistical Theory of Turbulence," *Proc 6th WSEAS Int. Conf. on Fluid Mechanics*, S. H. Sohrab et al., eds, WSEAS Press, Athens, pp. 165–172.

[9] Sohrab, S. H., 2007, "A Modified Theory of Turbulent Flow over a Flat Plate" *Proc 5th WSEAS Int. Conf. on Fluid Mechanics*, S. H. Sohrab et al., eds, WSEAS Press, Athens, pp. 165–172.

[10] Inkman, M. J., and Sohrab, S. H., 2008, "Comparisons between Velocity Profiles according to the Modified and the Navier–Stokes Equations of Motion and the Experimental Measurements for Laminar Boundary Layer over a Flat Plate", *Proc. 12th WSEAS Int. Conf. on Computing and Computational Techniques in Sciences*, J. Maza et al., eds., WSEAS Press, Athens, pp. 116–124.

[11] Burger, J. M., 1924, "The Motion of a Fluid in the Boundary Layer along a Plane Smooth Surface", *Proc. 1st Int. Cong for Applied Mechanics*, Delft.

[12] Burgers, J. M., and van der Hege Zynen, B. G., 1924, "Preliminary Measurements of the Distribution of the Velocity of a Fluid in the Immediate Neighbourhood of a Plane Smooth Surface", *Verhand. Akad. V. Wetesch*, Amsreada (iesectie) Deel XIII, No. 3.

[13] van der Higge Zjnen, B. G., *Measurements of the Velocity Distribution in the Boundary Layer along a Plane Surface*, Thesis Delft, 1924.

[14] Hansen, M., 1930, "Velocity Distribution in the Boundary Layer of a Submerged Plate", *NACA TM*, No. 585.

[15] Dhawan, S., 1953, "Direct Measurements of Skin Friction", *NACA*, TN 2567.

[16] Büttner, L. and Czarske, J., 2005, "Investigation of the Influence of Spatial Coherence of a Broad–area Laser Diode on the Interference Fringe System of a Mach–Zehnder Interferometer for Highly Spatially Resolved Velocity Measurements", *Applied Optics*, **44**, 9.

[17] Nikuradse, J., 1942, "Laminare Reibungsschichten an der längsangetrömten Platte. Monograph", *Zentrale f. wiss. Berichtswesen*, Berlin.

[18] Sohrab, S. H., 2004, "Scale–Invariant Forms of Conservation Equations in Reactive Fields and a Modified Hydro–Thermo–Diffusive Theory of Laminar Flames" *WSEAS Transactions on Mathematics*, Issue 4, **3**, 755.

[19] Bejan, A., *Convection Heat Transfer*, John Wiley & Sons Ltd, 4th Revised Edition, 2004.

[20] Bird, R. B., Stewart, W. E., and Lightfoot, E. N., *Transport Phenomena*, John Wiley & Sons Ltd, 2nd Edition, 2001.

[21] Probstein, R., *Physicochemical Hydrodynamics*, John Wiley & Sons; 2nd Edition, 2003.

[22] Arkeryd, L., 1972, "On the Boltzmann Equation Part II: The Full Initial Value Problem", *Arch. Rational Mech. Anal.* **45**, pp. 17–34.

[23] Batchelor, G. K., *An Introduction to Fluid Dynamics*, Cambridge University Press, 1967.

[24] Golse, F., 2011, "From the Kinetic Theory of Gases to Continuum Mechanics", *AIP Conf. Proc.* American Institute of Physics", **1333**, pp. 15–27.

[25] Sohrab, S. H., 2008, "A Modified Scale Invariant Statistical Theory of Turbulence", *Proc 6th WSEAS Int. Conf. on Fluid Mechanics*, S. H. Sohrab et al., eds, WSEAS Press, Athens, pp. 165–172.

[26] LeVeque, R., *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press, 2002.

[27] Toro, E. F., *Riemann Solvers and Numerical Methods for Fluid Dynamics*, Springer–Verlag, 1999.

[28] Patankar, S. V., *Numerical Heat Transfer and Fluid Flow*, Hemisphere, Washington, DC, 1980.

[29] Jasak, H., Error Analysis and Estimation for the Finite Volume Method with Application to Fluid Flows, Ph. D. Thesis, Imperial College, London, 1996.

[30] Hirsch, C., *Numerical Computation of Internal and External Flows*, John Wiley & Sons, 1991.

[31] Ferziger, J. H., and Peric, M., C*omputational Methods for Fluid Dynamics*, Springer–Verlag, 3rd Edition, 2002.

[32] OpenFOAM Wiki website.

[33] Donea, J., 1982 "An Arbitrary Lagrangian–Eulerian Finite Element Method for Transient Fluid–Structure Interactions, *Computer Methods in Applied Mechanics and Engineering*, 33, 689–723.

[34] Farhat, C., Geuzaine, P., and Grandmont, 2001, "The Discrete Geometric Conservation Law and the Nonlinear Stability of ALE Schemes for the Solution of Flow Problems on Moving Grids", *Journal of Computational Physics*, **174**, Issue 2, pp. 669–694.

[35] Farhat, C., Geuzaine, P., and Grandmont, C., 2001, "The Discrete Geometric Conservation Law and Its Effects on Nonlinear Stability and Accuracy", *15th AIAA Computational Fluid Dynamics Conference*, Anaheim, California, paper 2001–2607.

[36] Guillard, H., 2000, "On the Significance of the Geometric Conservation Law for Flow Computations on Moving Meshes", *Comput. Meth. Appl. Mech. Eng.*, **190**, pp. 1467.

[37] Jasak, H., *Dynamic Mesh Handling in OpenFOAM*, FSB, University of Zagreb, Croatia, Wikki Ltd, United Kingdom.

[38] NKonga, B., and Guillard, H., 1994, "Godunov Type Method on Non–structured Meshes for Three–dimensional Moving Boundary Problems", *Comp. Meth. Appl. Mech. Eng.*, **113**, pp. 183.

[39] Koobus, B., and Farhat, C., 1999, "Second–order Time–accurate and Geometrically Conservative Implicit Schemes for Flow Computations on Unstructured Dynamic Meshes", *Comp. Meth. Appl. Mech. Eng.*, **170**, pp. 103.

[40] Prandtl, L., 1904, "Über Flüssigkeitsbewegung bei sehr kleiner Reibung", *Verh. III. Intern. Math. Kongr.*, Heidelberg, 1904, S. 484–491, Teubner, Leipzig, 1905.

[41] Wan, B., Benra, F. K., and Sohrab, S. H., 2009, "A Comparative Numerical Study of the Modified versus the Navier–Stokes Equations of Motion", *Proc. 7th WSEAS Int. Conf. on Fluid Mechanics*, A. C. Benim et al., eds., WSEAS Press, Athens, pp. 157–162.

[42] Schultz–Grunow, F., 1940, "New Frictional Resistance Law for Smooth Plates", *Luftfahrtforschung*, **17**, No. 8, pp. 239–246.

[43] Thom, A., and Swart, P., 1940, "The Forces on an Aerofoil at Very Low Speeds", *Journal of the Royal Aeronautical Society*, **44**, pp. 761–770.

[44] Abbott, I. H., and von Doenhoff, A. E., *Theory of Wing Sections, including a Summary of Airfoil Data*, Dover Publication, New York, 1959.

[45] *OpenFOAM User Guide*, 2008.

[46] Taylor, A. M. K. P., Whitelaw, J. H., and Yianneskis, M., 1982, "Curved Ducts with Strong Secondary Motion: Velocity Measurements of Developing Laminar and Turbulent Flow," *Journal of Fluids Engineering*, **104**, pp. 350–359.

[47] Herrig, L. J., Emery, J. C. and Erwin, J. R., 1957, "Systematic Two–dimensional Cascade Tests of NACA–65 Series Compressor Blades at Low Speeds", *NACA*, TN 3916.

[48] Wang, W. S., *Experimental Study and Numerical Analysis of 3D Flow Field and Secondary Flow Mechanism in Annular Turbine Cascades*, Ph.D. Thesis, Institute of Engineering Thermophysics, Chinese Academy of Sciences, Beijing, 1992.

[49] Clancy. L.J., *Aerodynamics*, Pitman Publishing Ltd, London, 1975.

[50] Dring, R. P., Joslyn, H. D., Hardwin, L. W. and Wagner, J. H., 1982, "Turbine rotor–stator interaction", *ASME Journal of Engineering Power*, **104**, pp. 729–742.

[51] Lefcourt, M.D., 1965, "An Investigation into Unsteady Blade Forces in Turbomachines", *J. Eng. for Power Trans.* ASME, **87**, Ser. A: 345–354.

[52] Japikse, D., Marscher, W. D., and Raymond, B. F., *Centrifugal Pump Design and Performance*. Concepts TTI Inc., 1997.

[53] Ubaldi, M., Zunino, P. and Ghiglione, A., 1998, "Detailed flow measurements within the impeller and the vaneless diffuser of a centrifugal turbomachine", *Exp. Thermal and Fluid Sci.*, **17**, pp. 147–155.

[54] Feng, J. J., *Numerical and Experimental Investigations on Rotor–Stator Interaction in Radial Diffuser Pumps (Berichte Aus Dem Maschinenbau)*, Ph. D Dissertation, University of Duisburg-Essen, Shaker Verlag GmbH, 2009.

[55] http://openfoamwiki.net/index.php/Sig_Turbomachinery_/_ERCOFTAC_centrifugal_pump_with_a_vaned_diffuser.

[56] Combès, J. F., 1999, "Test Case U3: Centrifugal Pump with a Vaned Diffuser", *ERCOFTAC Seminar and Workshop on Turbomachinery Flow Prediction VII*, Aussois, pp. 4–7.

[57] Ubaldi, M., Zunino, P., Barigozzi, G. and Cattanei, A., 1996, "An Experimental Investigation of Stator Induced Unsteadiness on Centrifugal Impeller Outflow", *Journal of Turbomachinery*, **118**, pp. 41–54.

[58] Ubaldi, M., Zunino, P., Barigozzi, G. and Cattanei, A., 1996, "LDV Investigation of the Rotor–Stator Aerodynamic Interaction in a Centrifugal Turbomachine", *8th International Symposium on Applications of Laser Techniques to Fluid Mechanics*, Lisbon.

[59] Bert, P. F., Combès, J. F. and Kueny, J. L., 1996, "Unsteady Flow Calculation in a Centrifugal Pump Using a Finite Element Method", *Proceedings of the XVIII IAHR Symposium on Hydraulic Machinery and Cavitation*, pp. 371–380.

[60] Combès, J. F., Bert, P. F. and Kueny, J. L., 1997, "Numerical Investigation of the Rotor–Stator Interaction in a Centrifugal Pump Using a Finite Element Method", *Proceedings of the 1997 ASME Fluids Engineering Division Summer Meeting*, FEDSM97–3454.

[61] Torbergsen, E. and White, M. F., 1997, "Transient Simulation of Impeller/Diffuser Interactions", *Proceedings of the 1997 ASME Fluids Eng Division Summer Meeting*, FEDSM97–3453.

[62] Sato, K., *Blade Row Interaction in Radial Turbomachines,* Ph.D. Thesis, Durham University, 1999.

[63] He, L. and Sato, K., 1999, "Numerical Solution of Incompressible Unsteady Flows in Turbomachinery", *Proceedings of the 3rd ASME/JSME Joint Fluids Eng Conf*, FEDSM99–6871, San Francisco.

[64] Sato, K. and He, L., 2001, "Numerical Investigation into the Effects of a Radial Gap on Hydraulic Turbine Performance", *Proc. Instn Mech Engrs*, **215**, Part A, pp. 99–107.

[65] Page, M., Théroux, E. and Trépanier, J.–Y., 2004, "Unsteady Rotor–stator Analysis of a Francis Turbine", *22nd IAHR Symposium on Hydraulic Machinery and Systems*, Stockholm–Sweden.

[66] Page, M. and Beaudoin, M., 2007, "Adapting OpenFOAM for Turbomachinery Applications", *2nd OpenFOAM Workshop*, Zagreb.

[67] Petit, P., Page, M., Beaudoin, M., and Nilsson, H., 2009, "The ERCOFTAC Centrifugal Pump OpenFOAM Case–study", *3rd IAHR International Meeting of the Workgroup on Cavitation and Dynamic Problems in Hydraulic Machinery and Systems*, Brno, Czech Republic.

[68] Xie, S. S., *Studies of the ERCOFTAC Centrifugal Pump with OpenFOAM*, Master's Thesis, Chalmers University of Technology, 2010.

[69] M. Beaudoin and H. Jasak, 2008, "Development of a Generalized Grid Interface for Turbomachinery simulations with OpenFOAM", *Open Source CFD International Conference*.

Lebenslauf

Der Lebenslauf ist in der Online-Version aus Gründen des Datenschutzes nicht enthalten.